

La rivista per utenti di C-64/128 ed Amiga

COMMODORE GAZETTE

Novità per l'Amiga:

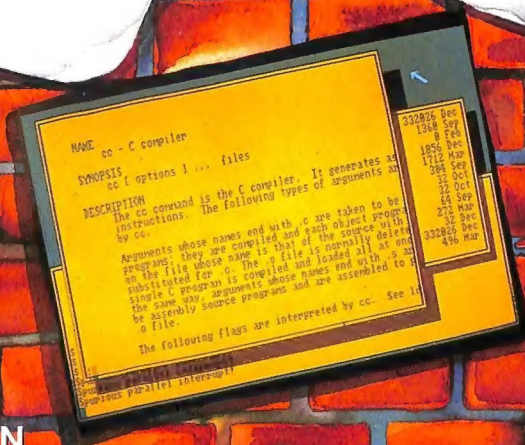
- UNIX E 68020
- A590, UN HARD DISK PER A500
- EMULATORE MACINTOSH
- STAMPANTE MPS 1224

Spec. Abb. post. Gr. III/70 - Lire 8.000



Software sotto accusa:

**VIDEOGIOCHI, NAZISMO
E VIOLENZA**



Programmare l'Amiga:

**AREXX, WShell E CONMAN
CYGNUS PROFESSIONAL
GADGET PROPORZIONALI
BASIC E GRAFICA AVANZATA**

Commodore 64 e 128:
RECENSIONI E LISTATI

IL PUNTO DI ARRIVO DELLE IDEE.

Come sempre le idee vincenti raggiungono l'obiettivo e noi della **Technitron Data** abbiamo messo ordine tra le tante proposte relative ai sistemi periferici per offrirvi solo le migliori. Noi della **Technitron Data** abbiamo l'esclusiva delle stampanti **OKI**, siamo distributori dei prodotti **Summagraphics** (digitalizzatori), garantendo oltre all'assistenza tecnica tutti i materiali di consumo e gli accessori. Noi del Gruppo **Technitron**, attualmente leader europeo nella distribuzione di questi prodotti, forti di tutta l'esperienza possibile delle esigenze degli utenti qualificati, siamo certi di avere la soluzione giusta per i vostri problemi: chiamateci!

02/90076410

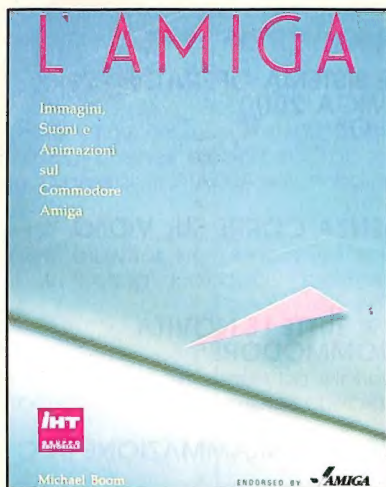
06/5042446

Technitron

DATA

dialogo continuo

Un computer chiamato AMIGA



L' AMIGA

Come il Commodore Amiga ha cambiato il mondo dei computer, così il volume *L'Amiga* proietterà la vostra immaginazione lungo nuovi orizzonti di creatività. Preparatevi a ottenere stupefacenti risultati dal vostro computer: sofisticate immagini video, suoni e musica, sequenze animate da registrare su videocassetta, e molto di più. In questo volume troverete inoltre consigli utili per ottenere il meglio dall'Amiga Basic e da prodotti software eccezionali come Deluxe Paint, Deluxe Music e Deluxe Video.

416 pagine, oltre 100 illustrazioni, L. 60.000

I L MANUALE DELL'AMIGADOS

Questo volume è l'unica documentazione ufficiale realizzata dalla Commodore sul sistema operativo dei computer Amiga. Contiene tre libri (Il Manuale per l'utente, Il Manuale per il programmatore e Il Manuale di riferimento tecnico) che costituiscono nel loro complesso la guida più completa per ogni utente dell'Amiga, dal principiante che vuole usare i comandi dell'AmigaDOS (che non sono spiegati dall'opuscolo in dotazione al computer), fino al programmatore evoluto che troverà utili informazioni per programmare in C e in Assembly.

376 pagine, L. 60.000



- 5** NOTE EDITORIALI
- 6** LA POSTA DELLA GAZETTE
- 9** SOFTWARE GALLERY
War in Middle Earth
Police Quest
Wec Le Mans
Blasteroids
The Legend of Blacksilver
Empire
Italy '90 Soccer
Pac-Land
Draw 2000
TV Sports Football
R-Type & Denaris
Zany Golf
Purple Saturn Day
Last Duel
- 35** SOFTWARE HELPLINE
Zak McKracken
and the Alien Mindbenders
- 39** CLASSIFICA SOFTWARE
- 40** INPUT/OUTPUT
- 120** COMMODORE NEWS
Novità dall'Italia e dall'Estero
- 124** CLASSIFIED
- 126** INDICE
DEGLI INSERZIONISTI
- 127** SERVIZIO LETTORI

COMMODORE GAZETTE

Una pubblicazione



Direttore Responsabile

Massimiliano M. Lisa

Redazione

Luca Giachino (capo servizio redazione tecnica)
Mauro Gaffo (redattore)
Dario Greggio (redattore tecnico)
Nicolò Fontana Rava (servizi speciali)

Collaborazione Editoriale

Fabio Rossi, Alfredo Macchi, Marco Menichelli,
Filippo Moretti, Dario Zanon

Corrispondenti USA

William S. Freilich (sezione sviluppo)
Daniela D. Freilich (coordinamento generale)

Collaborazione Editoriale USA

Louis R. Wallace, Ervin Bobo,
Eugene P. Mortimore, Harriet Maybeck Tolly

Inviato speciale USA

Matthew Leeds

Segretaria di Redazione

Giovanna Varia

Impaginazione e Grafica

Antonio Gaviraghi (capo servizio)
Andrea De Michelis

Fotografia

A.&G.,
Patricia Leeds

Disegni

Marco Piazza, Giuseppe Festino, Lara Parravicini

Direzione, Redazione, Amministrazione

IHT Gruppo Editoriale S.r.l.
Via Monte Napoleone, 9 - 20121 Milano

Fotocomposizione

IHT Gruppo Editoriale S.r.l. - Reparto grafica

Fotolito

C.S.T. S.r.l.
Via Melchiorre Gioia, 61 - 20124 Milano

Stampa

Eurograph S.p.A.
Via Oroboni, 38 - 20161 Milano

Distribuzione per l'Italia

Messaggerie Periodici S.p.A.
V.le Farnagosta, 75 - 20142 Milano
Tel. 02/8467545

Distribuzione per l'estero

A.I.E. S.p.A.
Via Gadames, 89 - 20151 Milano
Tel. 02/3012200

Pubblicità

IHT Gruppo Editoriale S.r.l.
Via Monte Napoleone, 9 - 20121 Milano
Ufficio Pubblicitario IHT
Italia ed Estero 02/794181-799492-792612-794122
Fax 02/784021 - Telex 334261 IHT I

Servizio Abbonamenti

IHT Gruppo Editoriale

Servizio Abbonati

Via Monte Napoleone, 9 - 20121 Milano

SEGRETERIA ABBONAMENTI

Linea per registrazione abbonamenti

02/794181-799492-792612-794122

Commodore Gazette

Costo abbonamenti

Italia

12 numeri L. 96.000

24 numeri L. 192.000

36 numeri L. 288.000

Estero:

Europa L. 150.000 (10 numeri)

Americhe, Asia... L. 200.000 (10 numeri)

I versamenti devono essere indirizzati a:

IHT Gruppo Editoriale S.r.l.

Via Monte Napoleone, 9 - 20121 Milano

mediante emissione di assegno bancario o vaglia postale

Arretrati

Ogni numero arretrato: L. 16.000 (sped. compresa)

Autorizzazione alla Pubblicazione

Tribunale di Milano n. 623 del 21/12/85

Periodico Mensile - Sped. in abb. post. gr. III/70

ISSN: 0394-6991

La IHT Gruppo Editoriale è iscritta nel Registro Nazionale della Stampa al n. 2148 vol. 22 foglio 377 in data 5/6/1987

Commodore Gazette è una pubblicazione

IHT Gruppo Editoriale

Copyright © 1989 by IHT Gruppo Editoriale S.r.l.

Tutti i diritti riservati.

Nessuna parte della rivista può essere in alcun modo riprodotta senza autorizzazione scritta della IHT Gruppo Editoriale. Manoscritti e foto originali, anche se non pubblicati, non si restituiscono.

I contributi editoriali (di qualunque forma), anche se non utilizzati, non si restituiscono.

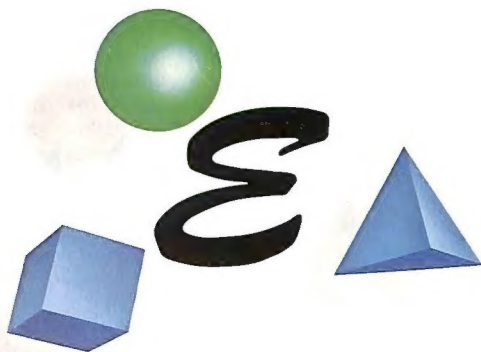
Non si assume alcuna responsabilità per eventuali errori od omissioni di qualsiasi tipo.

Commodore Gazette è un periodico indipendente non connesso in alcun modo con la Commodore Business Machines Inc. né con la Commodore Italiana S.p.A. PET, CBM, Vic-20, C-64, C-128, Amiga... sono marchi protetti della Commodore Business Machines. Nomi e marchi protetti sono citati senza indicare i relativi brevetti. Macintosh è un marchio della Apple, IBM PC AT, XT... sono marchi protetti dalla International Business Machines.

Associato
alla U.S.P.I.
(Unione Stampa
Periodica Italiana)



NOTE EDITORIALI



Esaminando la posta che riceviamo ogni giorno dai nostri lettori, abbiamo notato che nelle varie lettere ci vengono proposte con insistenza due domande: come mai *Commodore Gazette* non viene pubblicata con regolarità ogni mese? E perché, nonostante siano già stati annunciati da molto tempo, i volumi della IHT *Programmazione l'Amiga I e II* non sono ancora disponibili? Vista la quantità di lettere, e anche di telefonate, che riceviamo in merito, desideriamo una volta per tutte fare piena luce sull'argomento in modo da mantenere con i nostri lettori un rapporto di piena trasparenza.

Sicuramente per tutto il 1989 *Commodore Gazette* continuerà a uscire quasi sempre ogni due mesi. Non è stata inoltre ancora pianificata l'uscita su una regolare base mensile neanche per il 1990, e quindi probabilmente i tempi di pubblicazione rimarranno gli stessi anche per il nuovo anno. Questi tempi sono frutto di una decisione precisa: la Casa editrice della rivista ha degli standard qualitativi piuttosto elevati, il che ovviamente si traduce in costi di produzione a loro volta elevati. E per far fronte a questi costi su base mensile sarebbe necessario un gettito pubblicitario molto maggiore. Infatti, nonostante nell'ambito dell'informatica i potenziali inserzionisti siano numerosi, quelli specializzati in ambito Commodore sono una minoranza.

Gli introiti di *Commodore Gazette* sono quindi per la maggior parte quelli derivanti dalla vendita nelle edicole. E dal momento che queste vendite sono molto elevate - a tal punto da collocare il periodico tra le riviste d'informatica più vendute in Italia - *Commodore Gazette* continua a essere presente nelle edicole dal 1986.

La regolare mensilità sarebbe in ogni caso possibilissima: lo dimostrano le altre testate presenti in edicola dedicate ai computer Commodore. Ma per essere mensili sarebbe necessario ridurre i costi e quindi anche la qualità. E dal momento che gli standard qualitativi della Casa editrice impongono certi livelli, la periodicità di *Commodore Gazette* deve rimanere quella attuale. Almeno sino a quando non cambierà qualcosa nel panorama pubblicitario in ambito Commodore. I nostri

abbonati non hanno comunque nulla da temere. L'abbonamento è infatti legato al numero di fascicoli effettivamente ricevuti, e non ai mesi.

Per quel che riguarda i due volumi *Programmazione l'Amiga I e II*, va detto che hanno subito due tipi di ritardo. Il primo causato dal traduttore al quale era stato affidato il lavoro, che non solo ha superato di gran lunga i tempi massimi previsti, ma ha anche svolto un lavoro incompleto e non accurato; tanto da rendere necessario l'intervento di un nuovo traduttore.

Il secondo ritardo è invece legato ancora una volta agli standard qualitativi. I due volumi nella versione originale in lingua inglese contengono infatti un numero d'inesattezze superiore a quello tollerato dagli standard IHT. Si è così deciso d'intervenire sui due testi con un'ampia revisione tecnica. Questo vuole dire che i due libri usciranno parecchi mesi dopo la data prevista, ma conterranno tanti e tali aggiornamenti, migliorie e correzioni da giustificare ampiamente qualsiasi attesa. Il loro acquisto sarà quindi indispensabile anche per chi avesse nel frattempo comprato la versione in lingua inglese.

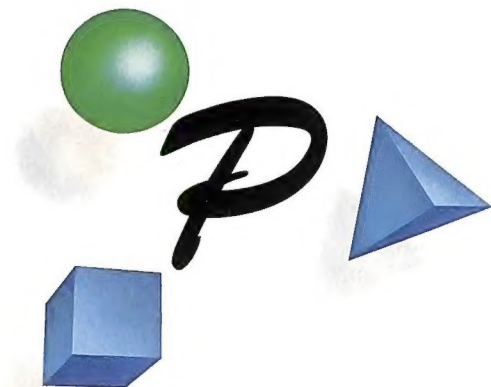
Le date di uscita previste sono fine luglio '89 per il volume II e fine ottobre '89 per il volume I. Il fatto che il volume II esca prima del volume I è dettato da ragioni di carattere tecnico. Questo non dev'essere comunque motivo di preoccupazione: pur integrandosi reciprocamente per coprire un vasto ventaglio di argomenti, i due libri sono indipendenti. Inoltre, nel corso del 1990 verranno pubblicati altri tre volumi che insieme ai due precedenti costituiranno una completa biblioteca per la programmazione dell'Amiga.

In definitiva, la politica editoriale della IHT Gruppo Editoriale è intimamente legata a standard qualitativi elevati. Talvolta questo si traduce in ritardi economicamente dannosi per la Casa editrice stessa. E nonostante la facilità con cui si potrebbe abbassare la qualità in favore di una produzione più rapida la IHT continuerà a rimanere fedele alla sua politica. Questo nella convinzione che i lettori italiani si meritino di più di quello che *altre* Case editrici offrono... molto di più.

ML

LA POSTA DELLA GAZETTE

LA VOCE DEI NOSTRI LETTORI



COMPETENZA E CORTESIA

Vorrei far sapere la mia opinione sulla scarsa accuratezza con cui la Casa madre sceglie i Commodore Point.

Alcuni giorni fa ho consultato l'elenco dei sopracitati punti di rivendita Commodore e ho notato (con molto piacere iniziale) che ne esisteva uno anche vicino alla mia abitazione, a Montecatini Terme (PT). Trattasi della ditta "Zanni & C." in Corso Roma 45 (non mi piace essere anonimo). Mi sono recato, dopo alcuni giorni, nel tale negozio e con molto dispiacere ho dovuto "beccarmi" la visione di una vetrina in cui c'era un bel Commodore Amiga 500 e un frullatore, per non parlare di altri microelettronici. Molto amareggiato dalla visione di un ennesimo negozio di elettrodomestici che si improvvisava rivenditore di computer, entro ugualmente. Ebbene, l'unica cosa riguardante la Commodore che c'era nel negozio era quell'esemplare di Amiga e alcuni giochi. Tutto il resto erano dischi (musicali), televisioni, materiale elettrico ecc. ecc.

Quello che voglio dire alla Commodore è che selezioni meglio i Commodore Point, e che

non dia questa "carica" a negozi di elettrodomestici dove si dice ai clienti che l'espansione A501 per l'Amiga sarà disponibile solo all'inizio dell'estate (non metto i punti esclamativi solo per non fare della retorica), come è successo a me a Montecatini. È una vergogna. Ripeto: è una vergogna che la Commodore commetta questi errori. È una cosa che fa perdere la fiducia degli utenti commodoriani nella Casa milanese.

A questo punto voglio fare un appello: voi utenti della Commodore che avete avuto un'esperienza come la mia, scrivete a *Commo-*

dore Gazette denunciando con nome e indirizzo i rivenditori che secondo voi non meritano di essere Commodore Point. Perché i Commodore Point devono essere dei punti di riferimento per gli utenti, che devono essere sicuri di trovarvi assistenza, competenza e cortesia.

In conclusione vorrei citare un esempio buono di Commodore Point: "Il Computer", Viale Colombo 216, Lido di Camaiore, dove personalmente ho potuto trovare competenza e serietà, qualità che dovrebbero avere tutti i Commodore Point.

Marco Magurno
Uzzano (PT)

**Indirizzate
tutta la corrispondenza
per la rivista a:**

COMMODORE GAZETTE
La posta della Gazette
Via Monte Napoleone, 9
20121 Milano

**Preghiamo i lettori di essere
concisi e concreti, per darci
modo di rispondere al
più grande numero possibile
di lettere. La redazione si
riserva comunque il diritto di
sintetizzare le lettere troppo
lunghe.**

PIRATI SOTTO ACCUSA

La pirateria è senza dubbio uno dei problemi più inquietanti nel panorama dell'informatica italiana, e ancora più inquietante è la disinformazione su tale argomento. Spesso e volentieri avete toccato questo problema, ma senza mai dedicargli uno spazio tutto suo per una trattazione sistematica.

Alcuni anni fa non avrei mai preso in considerazione l'idea di acquistare software originale a

causa del costo elevato e della difficile reperibilità dei prodotti, e quindi, come tanti altri, compravo ciò che il mercato nero offriva. Oggi le cose sono cambiate: il costo dei pacchetti software è calato e la distribuzione su larga scala si è dimostrata efficiente.

Eppure il fenomeno della pirateria è sempre presente. Le domande che mi sono poste sono queste: è possibile fare qualcosa contro tutte le persone che speculano illegalmente sul lavoro degli altri? A chi è preposto il compito di denunciare gli importatori abusivi e i negozianti che vendono software illegale? Nell'attesa di vedere nascere una rubrica dedicata alla pirateria, spero risponderete alle mie domande nello spazio dedicato ai lettori. Illustrando cosa può fare chi, come me, crede che esistano dei valori come l'onestà, e ha rispetto per il proprio lavoro così come per quello degli altri.

Stefano (NA)

Commodore Gazette ha dedicato alla pirateria un servizio speciale pubblicato nel n. 6187. Ci sembra che il nostro reportage sia la migliore risposta alle sue domande.

HARD DISK PER PC

Nel settembre 1988 ho acquistato un Commodore PC 10-III in configurazione base (640K e due drive da 360K). All'acquisto di questa macchina sono stato spinto in parte dal fatto che ho sempre posseduto macchine Commodore (VIC-20, C-64, C-128D) e in parte dalla lettura dell'esauriente articolo di prova pubblicato nel numero 3/88 di *Commodore Gazette*.

Il mio problema è recente, in quanto solo ora mi sono deciso a dotare la mia macchina di un hard disk 3,5" da 20 MB, e nei vari Commodore Point ai quali mi sono rivolto ho trovato totale

incompetenza. In un negozio ho avuto le risposte che cito: 1) utilizzando il controller esistente sul computer si può montare soltanto l'hard disk venduto dalla Commodore (quello posseduto di serie dal PC 20-III); 2) si può montare qualsiasi hard disk da 3,5" inserendo l'opportuno controller in uno degli slot disponibili.

Dato che dalla lettura dell'articolo mi sembra di capire che è possibile inserire qualsiasi hard disk XT compatibile provvisto d'interfaccia SCSI, gradirei sapere se la mia interpretazione è esatta e quindi se esiste la possibilità, servendosi del controller incluso nella piastra madre, di montare un qualsiasi hard disk da 3,5" XT compatibile purché con interfaccia SCSI, e in tale caso che modello e di quale marca (Seagate, Miniscribe, Rodime, Epson, Xebec).

Ruggero Baccetti
Scansano (GR)

Per il PC 10-III occorre un hard disk da 3,5" di tipo XT, o meglio dotato d'interfaccia XT. Si tratta di un tipo di hard disk ampiamente diffuso nel settore dei PC XT-compatibili, che oltre al disco e alla meccanica di accesso possiede una scheda logica di controllo ad alta integrazione. Per la loro compattezza, gli hard disk di tipo XT vengono impiegati specialmente quando lo spazio a disposizione non è molto, come nel caso del PC 10-III.

Per ora la Commodore non ha intenzione di distribuire singolarmente l'hard disk montato di serie nel PC 20-III, ma questo non è un problema, dal momento che gli hard disk di tipo XT sono praticamente tutti uguali, in termini di compatibilità hardware.

Sul mercato ve ne sono di molte marche e modelli. Sul suo PC potrebbe per esempio installare l'Epson HMD755, oppure il Western Digital WD93020.

I RICAMBI SCOMPARSI

Sono un utente deluso dalla qualità e dalla lentezza del servizio Commodore, in quanto il 15 gennaio ho portato a riparare il mio disk drive 1541 regolarmente garantito dalla Commodore in novembre e qual è stata la risposta del tecnico? «Mi dispiace, ma questo disk drive non può essere stato comprato in novembre». Lascio correre, e affido comunque il floppy alla VIGA che, dopo un mese di attesa, non ha ancora riparato la periferica per mancanza di pezzi... e alcuni drive sono fermi lì da agosto!

La Commodore si deve rendere conto che noi utenti siamo una forza da prendere in considerazione, altrimenti...

Francesco De Palo
Napoli

VELOCITÀ PER L'AMIGA

È la terza volta nel giro di poco tempo che vi scrivo, e questa volta vorrei porvi un quesito: quali reali vantaggi si hanno montando in un Amiga 500 un Motorola 68010 al posto del normale 68000? È conveniente o la differenza è minima? E per quanto riguarda il 68000 a 14 MHz, che prestazioni offre e quanto costa?

Vorrei anche informarvi che il metodo descritto da voi per utilizzare il Fast File System anche sui drive non funziona. Come mai? Cercate di rispondere al più presto, perché mi sono veramente stufato della lentezza del mio drive.

Luca Campana

Anzitutto, il Motorola 68010 è praticamente uguale al 68000: i due processori possiedono la stessa capacità d'indirizzamento, lo stesso bus dati da 16 bit, e lo stesso set d'istruzioni. L'unica differenza è

che il 68010 aggiunge tre registri, uno dei quali, il VBR, consente ai sistemi operativi di spostare la tavola dei vettori (256 vettori per le exception della CPU e di sistema) in qualunque posizione all'interno dello spazio indirizzabile, operazione che con il 68000 non è possibile (questa CPU considera infatti che la tavola dei vettori inizi sempre e solo dall'indirizzo zero dello spazio di memoria indirizzabile). Per un computer come l'Amiga, questo nuovo registro è pressoché inutile, e infatti è stato ideato per macchine in grado di avere in memoria contemporaneamente diversi sistemi operativi in esecuzione. Non ci sono quindi vantaggi a inserire un 68010 sul proprio Amiga, né in termini di velocità né per quanto riguarda set d'istruzioni o modi operativi più evoluti.

Per quanto riguarda il 68000 a 14 MHz, possiamo rispondere che ovviamente costa più del 68000 montato sull'Amiga, ma che purtroppo non sarà utilizzabile fino a quando non verrà realizzato un Amiga con un clock di 14 MHz.

Per altre idee su come aumentare le prestazioni del suo Amiga, la rimandiamo all'articolo "Schede acceleratrici per un Amiga turbo" apparso sul numero 6188 di Commodore Gazette.

A quanto ci risulta, seguendo le indicazioni di quell'articolo sull'impiego del Fast File System con i disk drive, tutto funziona perfettamente. Riprendiamo comunque qualche dettaglio per risolvere eventuali dubbi che potrebbero sorgere leggendo quell'articolo. Anzitutto occorre disporre del Fast File System, contenuto nella versione 1.3 del Workbench. Poi è necessario aggiungere nel file MountList, contenuto nella directory deus del disco sistema, i due gruppi di parametri illustrati nell'articolo, esattamente così come sono: il primo gruppo descrive un dispositivo logico di nome FFO: che consente di accedere a dischi formati con il Fast File System nel disk drive di serie, mentre il secondo si riferisce al dispositivo logico FF1;

relativo al primo disk drive aggiuntivo. Occorre precisare che se si dispone di un disk drive aggiuntivo è opportuno sperimentare questo sistema inserendo nella MountList soltanto il gruppo di parametri relativi a FF1.

Compiute queste operazioni, occorre digitare i comandi MOUNT FFO: e MOUNT FF1: per attivare i dispositivi logici, e il gioco è fatto. Per collaudare il tutto, supponendo di aver attivato FF1, basta inserire un disco da formattare nel disk drive aggiuntivo, che ora si chiama FF1:, e digitare il comando CHANGEDISK FF1: seguito dal comando SYS:SYSTEM/FORMAT DRIVE FF1: NAME "Disco FFS" NOICONS FFS. A questo punto disponiamo di un floppy disk formattato dal Fast File System, al quale si può accedere inserendolo nel disk drive aggiuntivo e riferendosi a quest'ultimo con il nome FF1:. Nel disk drive aggiuntivo si possono comunque inserire normali dischi AmigaDOS, purché vengano indicati con il consueto DF1:.

QUESITI SU FLIGHT SIMULATOR

Sono un felice possessore di una versione su cassetta per C-64 di Flight Simulator notevolmente ridotta; infatti si limita alla sola area di Chicago. Vorrei rivolgere alcune domande in proposito.

1) È forse realistico che col reliability factor a 50, in cinque minuti scarsi cinque strumenti (radiofaro 1, altimetro, tachimetro digitale, variometro e indicatore di velocità) siano oggetto di malfunzionamento?

2) Perché le piste uniche come Frasca Field di notte sono illuminate nelle strisce di mezzera, quando invece dovrebbero essere illuminati i contorni delle piste?

3) Che differenza c'è, ai fini della sicurezza, tra un decollo compiuto senza agire sulla manopola, col trimmer su di quattro

tacche e i flap a 10°, e un comune decollo con due tacche su e sempre 10° di flap?

Valerio Oddo
Napoli

1) Effettivamente, se si supera 50 con il reliability factor (0-100) si viene colti da un'incredibile scalogna che rende il volo quasi impossibile. Valori inferiori, comunque, garantiscono voli sempre avventurosi e costellati di ragionevoli imprevisti, e 50 livelli di sfortuna dovrebbero bastare a chiunque.

2) Per quanto riguarda il realismo della simulazione ambientale, ci sembra che lei pretenda un po' troppo. Sul C-64 Flight Simulator II fa già veri e propri miracoli, e anche se le piste non sono illuminate come nella realtà, non ce la sentiamo di metterlo sotto accusa.

3) Sia chiaro che prima di un decollo il trimmer dev'essere regolato in maniera che il timone di profondità si trovi in posizione neutra quando la cloche (che Lei chiama manopola) viene lasciata andare. Detto questo, passiamo alla sua domanda: i due tipi di decollo citati presentano un'importante differenza ai fini della sicurezza. Infatti, dando due tacche di trimmer e spingendo l'aereo al massimo con 10° di flap, metà della pista 18 dell'aeroporto Meigs Field di Chicago non è sufficiente per decollare e il Piper Archer II si trasforma in un idrovolante... Ma anche se ci si prende l'intera pista per decollare, due tacche di trimmer sono appena sufficienti. Quindi, se si desidera decollare in tutta tranquillità e lo spazio non è molto, quattro tacche di trimmer sono senz'altro raccomandabili.

Considerando il suo interesse per la simulazione del volo con Flight Simulator II, le raccomandiamo un istruttore davvero in gamba: il libro Co-Pilot, edito dalla IHT Gruppo Editoriale, un copilota indispensabile per apprezzare al meglio il fascino del volo elettronico. ■

SOFTWARE GALLERY

UNA GUIDA PER ORIENTARSI NEL MONDO DEL SOFTWARE



WAR IN MIDDLE EARTH



Un ottimo war game che ripercorre fedelmente l'intera saga del Signore degli anelli

Computer: C-64/128/Amiga
Supporto: Cassetta/Disco
Prezzo: L. 18.000/25.000/49.000
Produzione: Melbourne House
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago - 0332/212255)

Tutti gli appassionati di letteratura fantasy conoscono il nome di J. R. R. Tolkien, il grande scrittore inglese nato in Sudafrica, che con il suo grandioso ciclo della Terra di mezzo (*Lo Hobbit, Il signore degli anelli, Il Silmarillon*) ha creato un genere completamente nuovo e ha ispirato numerosissimi imitatori.

Qualsiasi libro, film, giocattolo o videogame fantasy prodotto oggi, si può fare risalire alla saga del geniale insegnante di filologia di Oxford o alle sue opere minori, ed è questo che ha convinto la Melbourne House ad acquistare tutti i diritti dei personaggi di Tolkien per la loro trasposizione nel mondo digitale.

Ogni giocatore di adventure ricorda con nostalgia lo storico *The Hobbit*, ispirato al libro omonimo, che per la prima volta offriva un parser di buon livello, o le due avventure del ciclo del *Signore degli anelli*, forse eccessi-

vamente complesse. Con *War in Middle Earth* la Melbourne House abbandona il terreno delle adventure per cambiare completamente genere. *War in Middle Earth* è infatti un war game che riproduce le vicende narrate nella trilogia del *Signore degli anelli*.

Anche se la trama della famosa trilogia di Tolkien è nota a molti, una breve introduzione sarà utile per comprendere più chiaramente la descrizione del gioco.

La storia comincia quando il piccolo hobbit Frodo riceve in eredità da Bilbo Baggins (il protagonista di *The Hobbit*) un anello magico: si tratta del Primo anello, che dona un singolare potere a chi lo possiede: l'invisibilità. Ma il Primo anello ha anche la pericolosa caratteristica di assorbire le energie vitali dei comuni mortali, mentre un abile stregone può utilizzarlo per dominare le forze del Male.

Questa è la ragione per cui il potente Sauron sta cercando di recuperare l'anello che gli venne rubato da Isildur. E Sauron comanda un gran numero di truppe di uomini e orchetti che, aiutati dai mostruosi demoni Nazgul, stanno percorrendo in lungo e in largo tutta la Terra di mezzo per recuperare il Primo anello.

SCHEDA CRITICA



INSUFFICIENTE

Un pessimo prodotto che non merita nessuna considerazione



MEDIOCRE

Il programma ha alcuni difetti di fondo anche se nel complesso raggiunge quasi la sufficienza.



DISCRETO

Un prodotto accettabile ma non aspettatevi grandissime emozioni



BUONO

Raccomandato vivamente tra i migliori programmi della sua categoria



OTTIMO

Eccezionale! Fino ad oggi non si era mai visto niente del genere

Frodo decide sotto consiglio del mago Gandalf di eliminare definitivamente Sauron portando a termine un'impresa rischiosissima: la distruzione dell'anello nelle fiamme vulcaniche di monte Fato, che purtroppo è anche il quartier generale di Sauron! Alla partenza per monte Fato sono presenti Frodo, lo stregone Gandalf il Grigio e altri tre hobbit, ma presto la Compagnia dell'anello accoglie nuovi membri fra cui l'elfo Legolas e il prode Aragorn, erede di Isildur. Nel suo viaggio verso sud il gruppo di eroi fa numerosi incontri e supera molti

una vicenda tanto complessa; diciamo subito che il giocatore controlla contemporaneamente tutti i personaggi non affiliati a Sauron o a Saruman, e che l'azione si svolge su tre schermate ben distinte.

La prima immagine che appare è quella di una mappa della Terra di mezzo, che riproduce esattamente quella contenuta nella confezione del gioco. Sulla mappa dello schermo ci si può spostare azionando un cursore a forma di mano guantata.

Premendo il pulsante del joystick, la mano si trasforma in una

finestra informativa dalla quale otteniamo informazioni dettagliate sull'identità e le condizioni dei personaggi. Una leggera pressione sul pulsante di azione ci permette di assegnare una destinazione al gruppo, di dividerlo o di unirlo a un altro, mentre un ultimo comando offre la possibilità di formare vere e proprie carovane. Una volta che ogni gruppo ha ricevuto i suoi ordini, il gioco torna sulla mappa principale.

Su questa schermata in alta risoluzione abbiamo a disposizione solo tre comandi: il primo serve per salvare la situazione su disco o per caricare un gioco precedentemente salvato, il secondo permette di leggere messaggi particolari e il terzo fa partire lo scorrere del tempo, facendo di conseguenza spostare le nostre truppe e quelle degli avversari.

Quando due fazioni opposte si scontrano, ecco apparire automaticamente il terzo e ultimo schermo di *War in Middle Earth*. La schermata riguardante i combattimenti è una delle più caotiche che si siano mai viste, e riesce a ricreare in maniera molto convincente la frenesia di una battaglia: su di un fondale sostanzialmente neutro, decine e decine di uomini e mostri si scontrano in un clangore metallico di spade e lance senza lasciarci un attimo per esaminare la situazione. Tutto quello che possiamo fare per influire sulle sorti della battaglia è spostare un cursore sul campo di gioco e selezionare un personaggio da controllare personalmente.

Le battaglie lasciano poco spazio per l'intervento umano. È quindi basilare l'aspetto strategico, ovvero gli spostamenti dei personaggi.

Lo schema ora descritto si ripete sino alla sconfitta totale di una delle due fazioni o alla cattura dell'anello da parte di Sauron. In realtà questa possibilità è piuttosto



Una schermata tratta dalla versione per l'Amiga di *War in Middle Earth*

pericoli, cercando di non farsi travolgere dalla guerra che infuria nella Terra di mezzo.

Come se non bastasse, oltre alle truppe di Sauron sono sulle tracce della Compagnia anche le forze di Saruman, che si fregia del titolo di aiutante di Sauron, ma fa di tutto per acquisire un potere superiore a quello del mago nero di monte Fato.

Versione per il C-64. Arriviamo finalmente agli aspetti pratici di *War in Middle Earth*. Per prima cosa si nota la semplicità di controllo, un fattore determinante in

lente d'ingrandimento che si sposta su di una riproduzione in scala più dettagliata della mappa principale: è qui che il giocatore si muove per la maggior parte del tempo, dal momento che solo in questa visuale è possibile identificare città, palazzi, torri e altri luoghi, ma soprattutto i personaggi del gioco. Questi ultimi vengono indicati per mezzo di scudi colorati, che possono rappresentare sia un unico personaggio sia un'intera armata.

Portando la lente d'ingrandimento su uno scudo, appare una

sto remota, dal momento che in battaglia il personaggio in possesso dell'anello può rendersi invisibile e fuggire dal luogo del pericolo, rimettendoci soltanto parte della sua resistenza fisica.

Versione per l'Amiga. Anche se lo schema di gioco si mantiene per lo più sulla linea descritta, la versione a 16 bit di *War in Middle Earth* ha ovviamente una propria personalità.

Oltre alla migliore definizione, un aspetto che era facilmente prevedibile, la versione per l'Amiga offre un'azzeccatissima musica introduttiva e un curioso linguaggio medioevaleggiante per i messaggi di sistema. Inoltre, vi sono due differenze "visive" che rendono abbastanza diverso l'aspetto grafico delle due versioni.

La prima differenza riguarda la possibilità di esaminare l'azione con la lente d'ingrandimento: questa opzione non si limita più a darci un rapporto scritto sulla locazione in esame, ma ce ne fornisce persino un'immagine digitalizzata. Se sotto la lente si trova un personaggio del gioco, appare un'animazione che ce lo mostra in viaggio.

Quanto ai combattimenti, osservandoli attraverso la lente d'ingrandimento appaiono in tutto lo splendore della grafica dell'Amiga, ma è a questo punto che si nota la seconda differenza. Le confusionarie e realistiche battaglie del C-64, che pure sarebbero risultate tanto più spettacolari sull'Amiga, sono state ridotte al formato delle schermate "di viaggio", e vengono combattute da un numero molto ridotto di contendenti. Essendo cambiata la disposizione grafica, anche il metodo di controllo si è adeguato alla nuova situazione: a ogni battaglia (che comunque possiamo ignorare e far combattere automaticamente dal computer) appare sullo schermo una finestra in cui vengono elencati i componenti delle nostre forze, ciascuno dei quali

affiancato da una serie di caselle.

Ogni casella corrisponde a un diverso comportamento sul campo: è possibile attaccare, caricare, retrocedere o fuggire, e ogni gruppo di armati ha la possibilità di muoversi indipendentemente dagli altri, rendendo per esempio possibile una carica da parte della cavalleria mentre i fanti retrocedono, e magari il comandante fugge verso lidi più sicuri.

War in Middle Earth è un ottimo programma in entrambe le versioni: che si giochi con

POLICE QUEST



Un graphic novel con un poliziotto come protagonista. Un ottimo testo rovinato da una pessima grafica

Computer: Amiga

Supporto: Disco

Prezzo: L. 49.000

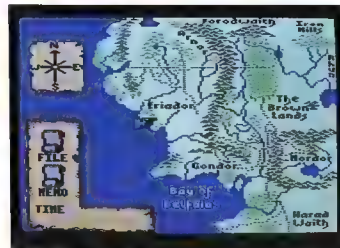
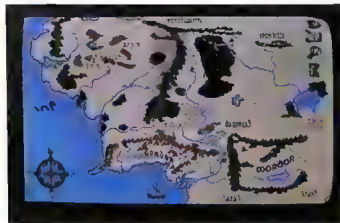
Produzione: Sierra On-Line

Distribuzione: Leader (Via Mazzini 15, 21020 Casciago - 0332/212255)

A avete mai pensato a come dev'essere la vera vita di un poliziotto?

Al di là dei cliché dei telefilm, gli uomini in divisa azzurra non passano certo le giornate scaricando quintali di proiettili su chi infrange la legge, né guidando le loro auto in folli inseguimenti che finiscono sempre per devastare bancarelle cariche di frutta. La vita di un tutore dell'ordine sarà fatta anche di azioni spettacolari, ma per la maggior parte del tempo passa in una monotona routine basata sulla scrupolosa osservanza dei regolamenti.

Questo almeno è quanto cercano di farci credere i programmatori della Sierra On-Line con il loro *Police Quest*, ultimo gioco della fortunata serie "Quest" a essere pubblicato in versione Amiga. La serie completa comprende una decina di titoli di ambientazione fantasy, spaziale o poliziesca, fra i più apprezzati nello striminzito campo ludico dei computer MS-DOS compatibili. La filosofia di *Police Quest* è quella dei graphic novel, di cui abbiamo parlato qualche tempo fa nella recensione di *Neuromancer*: l'ambiente del gioco cerca di simulare la realtà il più fedelmente possibile, facendo molto affidamento sulla grafica e limitando drasticamente l'intervento e le possibilità di controllo del giocatore.



La medesima schermata nella versione per l'Amiga (sopra) e per il C-64 (sotto)

l'ausilio di un microprocessore a 8 oppure a 16 bit, l'azione risulta sempre semplice da controllare e mantiene comunque un notevole livello di complessità.

Anche se i frequenti cambi di dischi richiesti con la versione a 16 bit possono disturbare il ritmo dell'azione, *War in Middle Earth* merita tranquillamente il titolo di miglior war game fantasy attualmente in circolazione, e occuperà ben presto un posto di rilievo nella collezione software di tutti gli amanti della strategia. **F.R.**

Come nella vita reale, il nostro rappresentante sullo schermo non è affatto il centro della narrazione: i personaggi del gioco hanno una loro vita propria, e gli avvenimenti si susseguono indipendentemente dalla nostra preparazione ad affrontarli. Uno dei nostri compiti principali è proprio quello d'inserirci nella complessa società del gioco.

Police Quest è ambientato a Lytton, una normalissima cittadina americana dove non manca nulla ma nessun elemento è eccezionale. Il parco pubblico non è il Central Park, la tangenziale non

tutto crimini connessi alla droga è in continuo aumento, e la polizia della città si trova sempre più sotto pressione per tentare di arginarli.

Questo è il momento in cui entriamo in scena noi, nei panni di un comune poliziotto in divisa. Per il momento dobbiamo adattare a svolgere diligentemente il lavoro di routine, sperando che i superiori notino la nostra buona volontà e ci diano la possibilità di dedicarci a compiti più importanti. Il primo giorno di gioco ci troviamo a pattugliare le strade in auto, ma la nostra meta è quella di

un 10-27, centrale. Chiedo un 11-41 e confermo 10-98»). Qualsiasi azione viene eseguita seguendo il Codice di pubblica sicurezza e una serie di procedure prestabilite, ma ciò non significa che il gioco sia noioso. Con il passare del tempo incontreremo sempre più imprevisti, e col susseguirsi delle promozioni avremo maggiori possibilità di prendere iniziative personali e di mettere alla prova la nostra inventiva.

La collaborazione gioca un ruolo molto importante in *Police Quest* e non è possibile alcun progresso senza un minimo di socializzazione con i nostri colleghi e con gli altri personaggi del gioco. Il nostro partner, in particolare, ricopre un ruolo molto importante nello svolgersi della trama, vista la sua disponibilità a rischiare la pelle senza pensarci troppo.

Police Quest è indubbiamente un programma enorme e molto curato, ma risente pesantemente delle sue origini. Le troppo mitizzate macchine IBM non sono certo i migliori ambienti di sviluppo per un gioco, e anche quando la velocità di esecuzione non è particolarmente importante (come in questo caso), le insulse capacità grafiche e sonore dei modelli di base costringono anche i migliori programmatori a dare una veste da VIC-20 o peggio alle loro creazioni.

Il fenomeno è particolarmente evidente in questo programma, il cui aspetto esteriore è tanto primitivo da risultare addirittura fastidioso: la grafica statica manca di realismo cromatico e prospettico, quella animata è tanto puerile da essere ridicola e il sonoro è limitato a qualche effetto cicaleggiante. Naturalmente la cosa non sarebbe così vergognosa se stessimo parlando della versione per IBM del gioco, ma il problema è che qui stiamo parlando dell'Amiga! È un vero shock vedere un prodotto che apparirebbe primi-



ha centinaia di corsie e l'ospedale non è nemmeno paragonabile ai colossali "Medical Center" dei telefilm. La vita della città scorre secondo ritmi assolutamente realistici, con una tranquillità di base rotta di quando in quando da piccoli problemi.

Fra gli incidenti stradali, le multe contestate e le risse in birreria, qualcosa però preoccupa sempre più l'opinione pubblica: un criminale, battezzato dai giornali "l'angelo della morte", ha fatto di Lytton la sua base operativa, e la sua presenza costituisce un incentivo alle attività criminali. Il numero di furti, rapine e soprat-

poter lavorare in borghese come infiltrati nell'organizzazione dell'angelo della morte, per arrestarlo ed estirpare il male alla radice, riportando Lytton all'usuale tranquillità.

La trama di *Police Quest*, scritta da un vero agente di polizia, è studiata per essere quanto più realistica possibile. L'esperienza sul campo dell'autore si fa sentire continuamente, come per esempio nell'obbligo di seguire una procedura di controllo dell'auto tutte le volte che si sale a bordo o nell'impiego di codici di comunicazione reali fra il centro radio della polizia e l'auto (i famosi «Ho

tivo su un C-64, fare la sua comparsa sullo schermo del computer che ha rivoluzionato il concetto di grafica e sonoro "personali".

Purtroppo anche l'interfaccia utente costituisce un grosso problema: il controllo del nostro poliziotto avviene infatti tramite una combinazione di joystick e input da tastiera che risulta poco pratica ma soprattutto troppo lenta, e che spesso conduce a fallimenti clamorosi indipendentemente dall'abilità del giocatore, specie nelle sequenze più movimentate. Una situazione simile è assolutamente inaccettabile in pieno 1989 e con una macchina come l'Amiga.

Chiudendo tutti e due gli occhi su queste pesantissime limitazioni, quello che rimane di *Police Quest* è una trama eccezionale, tanto realistica e dettagliata da poter essere definita una vera e propria simulazione. I personaggi, i luoghi e gli avvenimenti sono così ben caratterizzati da costituire uno dei migliori risultati ottenuti finora nel campo dei graphic novel. Si può affermare senza esitazioni che ci troviamo di fronte a un vero prodotto educativo. *Police Quest* è in grado d'insegnare veramente qualcosa sul funzionamento di un'organizzazione di polizia (benché il riferimento diretto sia alla polizia statunitense) e di dare importanti nozioni di educazione civica. Purtroppo questo realismo si paga, chissà perché, con una veste grafica orrenda e un sonoro inesistente, che non vengono controbilanciati nemmeno dalla ricca manualistica della confezione.

Il miglior consiglio che possiamo dare agli appassionati di questo genere di giochi è di provare - se ne avete la possibilità - il programma prima di una eventuale acquisto. *Police Quest* offre molte soddisfazioni e molte delusioni, in eguali proporzioni: colossali.

F.R.

WEC LE MANS



Un prodotto quasi amatoriale che non merita l'attenzione degli appassionati

Computer: C-64/128
Supporto: Cassetta/Disco
Prezzo: L. 18.000/21.000
Produzione: Imagine Software
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago - 0332/212255)

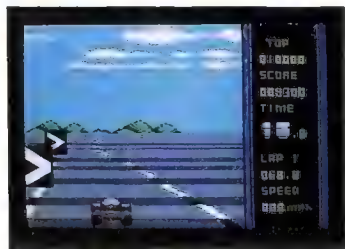
In un foglio multilingue (che dovrebbe rappresentare il manuale di gioco di *Wec Le Mans*) contenuto nella confezione del programma leggiamo: «La più estenuante e avvincente gara automobilistica del mondo è la ventiquattr'ore Wec Le Mans... un'esperienza unica nella gara di corsa più realistica e appassionante che ci sia».

Parole belle e convincenti, che fanno pensare immediatamente a un buon gioco di simulazione automobilistica, e che creano una sensazione di suspense durante il caricamento del gioco. Purtroppo però questa attesa viene gravemente delusa non appena ci si rende conto di trovarsi di fronte a una pessima rivisitazione dell'ormai obsoleto schema di gioco che, dal famoso *Pole Position* (Atari) in poi, è stato sistematicamente ripreso in tutti i giochi di questo tipo. Come se non bastasse, il programma non è nemmeno una buona versione di un vecchio gioco: propone una grafica di medio livello, un sonoro quasi inesistente (la cosa più bella è un'inutile musichetta che accompagna la schermata iniziale), e una giocabilità paragonabile a quella di un word processor.

Lo schermo si presenta nella solita configurazione dell'automobile vista da dietro che mentre corre vede variare le dimensioni delle vetture e degli oggetti che le vengono incontro sulla pista, con

un effetto di pseudo tridimensionalità. "Pseudo" perché le dimensioni variano tanto bruscamente che le altre auto fino a un certo punto sembrano piuttosto lontane, e si avvicinano di colpo quando stiamo per sorpassarle. Gli oggetti (segnali, alberi, guardrail...) compaiono all'improvviso e non si possono evitare se si è troppo vicini al bordo della pista.

Altro punto debole del gioco sono i comandi: pur essendo attribuite al joystick le funzioni tradizionali (leva avanti accelera, indietro frena, destra e sinistra sposta la macchina e il pulsante cambia marcia), la risposta della vettura è troppo brusca (forse perché non si tratta di una Formula Uno ma di una vettura



Endurance) e crea seri problemi soprattutto nelle curve, rendendo il gioco più difficile in un modo molto poco divertente.

Infine la grafica, come abbiamo già detto, è rozza e poco precisa (a volte interi sprite scompaiono) e non aiuta certo a raggiungere l'atmosfera di tensione competitiva promessa all'inizio. Spiace condannare senza appello un prodotto che, nonostante tutto, rappresenta il lavoro di molte persone, ma non è onesto presentare novità scadenti in confezioni coloratissime per rifilare agli acquirenti prodotti di serie B. È un atteggiamento che va a discapito dell'intera industria del software... che di problemi ne ha già molti.

N.F.R.

BLASTEROIDS

Il glorioso *Asteroids* arriva sugli schermi degli home computer, in una versione parzialmente ispirata al film *The Last Starfighter*



Computer: C-64/128/Amiga
Supporto: Cassetta/disco
Prezzo: L. 18.000/18.000/29.000
Produzione: Image Works
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago - 0332/212255)

L'industria dei videogiochi, sempre in cerca di soggetti interessanti da proporre al suo vasto pubblico, ha sfruttato spesso il successo di un titolo particolare come trampolino di lancio per prodotti simili. L'esempio più evidente è dato dalla saga dei labirinti alla *Pac-Man* (di cui trovate tutti i particolari nella recensione di *Pac-Land* in questo stesso numero), ma negli ultimi tempi c'è stato un vero e proprio recupero dei videogiochi dei tempi pionieristici.

Pong è diventato *Starball*, *Breakout* - meglio conosciuto come "il muro" - ha generato il bellissimo *Arkanoid* con tutti i suoi cloni e il leggendario *Space Invaders* si è trasformato in *Return of the Invaders*. Poteva fare eccezione un grande classico come *Asteroids*, precursore di tutti i coin op dotati di monitor vettoriale?

La risposta è naturalmente no, e l'onere di creare un degno seguito a questo cult-game è andato alla Tengen, nuova divisione dell'americana Atari preposta alla produzione di giochi a gettone. Per quanto non specificato nelle istruzioni, *Blasteroids* sfrutta alcuni elementi del film *The Last Starfighter*, la cui sceneggiatura basata su un videogioco futuristico aveva indotto la Atari ad acquistarne i diritti per il mercato dei videogame cinque anni fa.

Gli aspetti tratti dal film si limitano all'impiego di una particolare "arma finale" chiamata Ripstar e a una parte della colonna sonora nella schermata dei titoli, ma *Blasteroids* ripropone anche i classici concetti di trasformabilità e di armamento progressivo che, uniti alla grande giocabilità del coin op originale, lo rendono un prodotto senz'altro interessante.

Blasteroids si svolge in quattro diverse galassie, selezionabili da una schermata iniziale, e corrispondenti ad altrettanti livelli di difficoltà: ogni galassia è composta da un certo numero di settori, che vanno liberati dalla minaccia degli asteroidi vaganti per poter considerare concluso con successo il gioco. Prima di entrare nel vivo, sulle schermate che rappresentano i singoli settori, viene presentata una mappa galattica che preannuncia ciò che ci attende in ognuno di essi: eloquenti simboli indicano la presenza di asteroidi di diversi tipi e di astronavi-sentinella, lasciando intuire la difficoltà relativa.

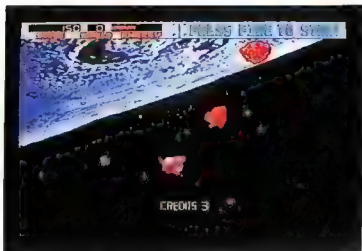
Una volta selezionato il settore, la nostra astronave viene proiettata in un quadro costituito da una schermata ovviamente di ambientazione spaziale, e al suono di un'ossessionante musica giocata completamente sui toni bassi veniamo assaliti da una miriade di rocce vaganti, che rotolano minacciose verso la nostra navicella.

I movimenti a nostra disposizione sono quelli resi famosi dal grande *Asteroids*: rotazione oraria e antioraria, e accelerazione nella direzione indicata dal muso del veicolo. Il pulsante di fuoco attiva come sempre il cannoncino di bordo, ma c'è un'importante caratteristica che differenzia *Blasteroids* dal suo fratello maggiore.

Tirando verso di noi il joystick, infatti, l'astronave si trasforma; in tutto abbiamo a disposizione tre veicoli. Il primo è una piccola navetta dotata di una notevole

capacità di accelerazione ma di resistenza e potenza di fuoco ridotte, utilissima nelle occasioni in cui è necessario inseguire un bonus particolare, fuggire da una situazione pericolosa o da un campo gravitazionale troppo potente. Utilizzare questo tipo di veicolo mette in luce i problemi relativi al controllo dei movimenti degli oggetti che attraversano lo spazio di *Blasteroids*: usare per troppo tempo i motori della nave ce ne fa perdere il controllo, proprio come accadeva nel gioco originale.

La seconda trasformazione a nostra disposizione converte la navicella in una vera e propria fortezza spaziale, il cui enorme volume è giustificato dalla resi-



Blasteroids nella versione per l'Amiga

stenza degli scudi della nave, ma che richiede pesanti contropartite in termini di rapidità di spostamento e di volume di fuoco. L'uso di questo mezzo è ovviamente consigliabile in situazioni d'immediato pericolo, come per esempio quando ci si trova assediati dalle misteriose navi spaziali che facevano una comparsa marginale anche nel vecchio *Asteroids*: questi avversari hanno la pessima abitudine di precipitarsi sulla nostra astronave cercando di eliminarci sia con i loro cannoni di bordo sia travolgendoci come kamikaze e assorbendo parte della nostra energia (indicata da una barra colorata nella parte alta dello schermo). Dalla lunghezza

della barra dipende la nostra resistenza, e quando l'indicatore giunge a zero l'unica speranza di sopravvivenza sta nell'eventuale presenza di crediti da convertire in altre riserve di energia, proprio come accade in sala giochi.

Il terzo veicolo è il cosiddetto "Warrior", dotato di grande potenza di fuoco ma di scarsa resistenza agli attacchi e di mobilità limitata. È probabile che l'uso di questo veicolo ci faccia completare in tempi ridotti i vari settori, ma è praticamente certo che nell'impresa verremo prosciugati di buona parte dell'energia a nostra disposizione: nasce quindi il problema di come reintegrare le scorte energetiche, che è stato risolto brillantemente.



La minor definizione della versione C-64

Alcuni asteroidi, riconoscibili per il loro colore rosso, quando vengono colpiti non si limitano a frammentarsi in parti sempre più piccole come è regola comune, ma rilasciano alcuni cristalli luminosi nel momento in cui vengono vaporizzati completamente. Inseguendo e raccogliendo questi cristalli recuperiamo parte della nostra energia, e riusciamo a prolungare la nostra esistenza digitale. Secondo lo stesso principio possiamo migliorare le caratteristiche della nostra nave distruggendo le astronavi avversarie; molte di loro, infatti, lasciano tra i relitti alcuni moduli che possiamo integrare nella nostra struttura semplicemente passandovi so-

pra. I moduli hanno di volta in volta l'effetto di aumentare le nostre capacità di difesa (scudi), di movimento (acceleratori) e di attacco (cannoni binari), ma possono anche fornire capacità meno ovvie: un apparecchio incrementa il potere perforante dei nostri colpi, un altro attira automaticamente i cristalli energetici verso la navicella e un altro ancora ci fornisce la favolosa Ripstar!

La Ripstar è un'arma che può essere utilizzata una sola volta per settore, e viene attivata con una combinazione di movimenti del joystick e del pulsante di fuoco. Ha l'effetto di far ruotare vorticosamente la navicella scatenando nel frattempo la massima potenza di fuoco dei cannoni, che riempiono così lo spazio attorno al nostro veicolo di proiettili che frantumano ogni altro elemento sullo schermo.

Procedendo nel gioco, agli asteroidi normali e a quelli rossi si aggiungono altre forme di pericolo spaziale: rocce bluastre che generano nuovi asteroidi finché non vengono fermate a suon di laser, "sanguisughe energetiche" che assorbono la nostra energia e così via. A lungo andare risulta evidente che non è possibile affrontare da soli tutti gli ostacoli. Proprio in vista di questa necessità, *Blasteroids* offre un'opzione che permette a due giocatori di muoversi sullo stesso schermo, senza limitare nessuna delle possibilità descritte precedentemente e al contrario aggiungendone di nuove.

Ci riferiamo soprattutto alla possibilità di unire in un unico veicolo le due astronavi, che si tramutano così in un terribile mezzo di distruzione. In questa configurazione, una delle astronavi funge da modulo principale corazzato preposto alla navigazione e l'altra da "torretta" rotante dotata di un'incredibile potenza di fuoco: con una simile accoppiata arrivare a ripulire l'intera ga-

lattia è decisamente più semplice, ma le sfide non sono ancora finite.

Una volta eliminata la minaccia degli asteroidi anche dall'ultimo settore, infatti, resta da affrontare il terribile e viscidissimo alieno Mukor, misterioso complice delle meteore vaganti. Questa orrenda bestia spaziale ha la forma di un gigantesco planetoido dotato di tentacoli che vanno eliminati uno per uno a colpi di cannone. Combattere una creatura grande quanto un quarto di schermo che sputa proiettili ininterrottamente e cerca d'investire la nostra piccola navetta non è un'impresa facile, ma per la liberazione della galassia vale la pena di correre il rischio.

Le versioni di *Blasteroids* per C-64 e Amiga non si distinguono se non per due aspetti, entrambi ovvi: l'elevata capacità di calcolo dell'Amiga, e la presenza di chip grafici di cinque anni più aggiornati rispetto a quelli del C-64. Ne risulta una grafica dotata di maggior definizione e una maggiore velocità (che in alcuni casi diventa addirittura un handicap). In entrambe le versioni, comunque, *Blasteroids* possiede un'elevatissima giocabilità. Considerando anche l'esistenza di quattro livelli, è prevedibile per questo programma una notevole longevità, per quanto il tema e la meccanica di gioco siano un tantino antiquati e ripetitivi.

La Image Works ha fatto veramente un ottimo lavoro di conversione del gioco Atari/Tengen (praticamente perfetto nel caso della versione a 16 bit), al punto che gli utenti Amiga possono permettersi di decidere se sono interessati all'acquisto semplicemente osservando *Blasteroids* in una sala giochi o in un bar. L'unico appunto che possiamo fare riguarda una certa ripetitività, ma anche questo piccolo difetto diventa trascurabile nel caso della partita a due. **F.R.**

THE LEGEND OF BLACKSILVER



Un gioco di ruolo un po' troppo semplice, chiaramente ispirato a Questron

Computer: C-64/128

Supporto: Disco

Prezzo: L. 18.000

Produzione: Epyx

Distribuzione: Leader (Via Mazzini 15, 21020 Casciago - 0332/212255)

Prima d'iniziare la recensione, va ricordato che la Epyx, la casa produttrice di questo programma, fu nel 1983 la prima software house che inserì nel proprio catalogo un gioco di ruolo, inaugurando quello che si rivelò poi un genere di notevole successo. Lo storico evento, lo ricordiamo per i nuovi arrivati, si ebbe con la prima versione (in Basic) di *Temple of Apshai*.

La realizzazione di *The Legend of Blacksilver* è stata affidata a un team di programmatori indipendenti capitanato dai famosi fratelli Dougherty (autori della saga di *Questron*), che hanno trasmesso al gioco il loro stile caratteristico.

Le premesse non sono particolarmente originali. Il giocatore assume l'identità di un umile accatone che, scelto durante uno strano sogno dalla principessa del regno di Thalen, si trova costretto ad accollarsi il compito di salvare le terre in cui egli stesso vive. Thalen ha una storia complessa e piuttosto tragica: come possiamo leggere nelle Cronache contenute nella confezione, infatti, il pianeta di Bantross (sul quale si trova Thalen) possiede una curiosa ricchezza che ha influenzato tutta la sua storia.

Nelle profondità di Bantross si trovano infatti ricchi filoni del rarissimo minerale chiamato

Blacksilver, che ha il potere di fungere da catalizzatore per gli incantesimi e i rituali negromantici eseguiti quotidianamente dai molti stregoni del pianeta.

Vista l'importanza del Blacksilver, nel passato sono scoppiate numerose guerre per il possesso dei filoni. Ma dopo la Guerra nera degli stregoni, le guerre sono terminate definitivamente: infatti, le isole di Maelbane si sono inabissate con i loro ricchissimi filoni di metallo magico.

Dopo un periodo di relativa pace, il barone Taragas decide di

del pianeta è affidata al potere del minerale). Aylea e il mago di corte Seravol decidono di trovare un avventuriero che liberi Durek e che sconfigga Taragas, e la scelta cade proprio su di voi...

La prima impressione, esaminando *The Legend of Blacksilver*, è quella di trovarsi di fronte a una nuova versione di *Questron*. Come nei precedenti giochi dei Dougherty, lo schermo è diviso in tre finestre principali nelle quali si svolge tutta l'avventura.

La finestra verticale sulla sinistra contiene tutti i comandi a



cercare il Blacksilver nei suoi possedimenti sul continente, e riesce a estrarre il prezioso minerale pochi anni dopo l'incoronazione di re Durek, padre della principessa Aylea. Forte dei nuovi poteri acquisiti grazie al possesso del metallo magico, Taragas rapisce re Durek e diventa il tiranno del maggiore continente di Thalen.

La situazione si fa sempre più critica, anche a causa delle catastrofi naturali provocate dalla scomparsa del Blacksilver dalle profondità di Bantross (la stabilità

nostra disposizione: con l'ausilio del joystick possiamo muovere un cursore luminoso e selezionare quello desiderato. Nella finestra orizzontale, in basso, appaiono i messaggi del gioco: è qui che possiamo leggere i dati sui combattimenti che affronteremo, i dialoghi intavolati con i vari personaggi e le descrizioni più complesse.

L'ultima finestra, la più grande, si trova sulla destra e contiene una rappresentazione del mondo che circonda il nostro personaggio. Questa può variare da una

visuale bidimensionale a volo d'uccello (nelle sequenze all'aperto e nelle città) a una grafica tridimensionale in soggettiva che appare quando penetriamo in alcune sezioni particolari del gioco, come negli immancabili dungeon o nelle cripte del sapere.

La grafica è leggermente migliorata rispetto a quella utilizzata in *Questron II*, ma rimangono le rappresentazioni "alla VIC-20" degli interni e dei personaggi, che abbassano notevolmente la qualità media della grafica del gioco. Un buon lavoro è stato invece

nostro personaggio è definito da una serie di attributi come Forza, Resistenza e così via, ma si scopre ben presto che gli attributi non variano con l'aumentare dell'esperienza acquisita: le proprie qualità si migliorano semplicemente superando una serie di veri e propri giochi di abilità che possono essere affrontati in qualsiasi momento. Se l'esito è negativo gli attributi peggiorano drasticamente.

The Legend of Blacksilver chiede poi al giocatore di acquisire alcune particolari "abilità speciali"

offre una vasta gamma d'impieghi, dal lavapiatti al contabile, che vengono regolarmente retribuiti permettendovi di rivestire il noioso ma originale ruolo di primo "avventuriero impiegatizio" della storia.

Una parola va spesa per la mappa, stampata su un grande foglio di carta patinata. Questa mappa di Thalen viene fornita insieme con una serie di etichette adesive, sulle quali sono indicati i nomi di tutti i luoghi del regno d'interesse per il giocatore, che vanno appiccate sul foglio una volta identificata la loro posizione esatta. Questa scelta, originale ma ben poco brillante, conduce in breve tempo a mappe impossibili da piegare ed etichette volanti, e non facilita certo un buon rapporto con il gioco. Forse è per questo che la Epyx offre nella confezione una cartolina con la quale ordinare un Hint book che fornisce la soluzione del gioco, che comunque è piuttosto semplice, soprattutto se la paragoniamo a quella di alcuni recenti giochi di ruolo come *Wasteland* o il colossale *Ultima V*.

The Legend of Blacksilver ha il suo più grande difetto nell'interfaccia utente, che pur essendo comodissima sacrifica alla giocabilità quella complessità che non dovrebbe mai mancare in un gioco di ruolo. Poter controllare tutto il gioco via joystick è effettivamente rilassante, ma implica anche combattimenti monotoni e molto poco spettacolari, interazione quasi nulla con i personaggi e un uso limitatissimo della magia.

Come tutti i prodotti della serie *Questron* (pur non appartenendo ufficialmente a questa categoria), il nuovo gioco della Epyx farà sicuramente la gioia dei meno esperti e dei più giovani, ma va sconsigliato vivamente a chi in un gioco di ruolo cerca trame complesse e mondi credibili.

F.R.



fatto nel campo degli effetti sonori: è possibile persino sentire il frangersi delle onde sulla spiaggia.

La struttura del gioco, come sempre, si discosta abbastanza nettamente da quella degli altri giochi di ruolo in circolazione. A parte il fatto che il nostro eroe ha la pessima abitudine di muoversi solitario sulle pericolose terre di Thalen, altri aspetti distinguono *The Legend of Blacksilver* dalla folla di eredi di *Dungeons & Dragons*.

Come tutti gli eroi fantasy, il

che risultano indispensabili per superare determinate situazioni. Ebbene, per la prima volta nella storia dei giochi di ruolo queste abilità non si ottengono attraverso sanguinari combattimenti, ma grazie a un lungo e paziente lavoro di biblioteca, andando a consultare negli intricati sotterranei polverosi volumi contenuti in apposite teche chiuse a chiave.

Un'ultima particolarità è la possibilità di lavorare per guadagnarsi il pane, piuttosto che vagare lungo terre desolate massacrando decine di mostri. Il gioco

EMPIRE



La quarta puntata di Starfleet abbandona l'ambientazione spaziale e ci guida alla conquista del pianeta Arabella

Computer: Amiga
Supporto: Disco
Prezzo: L. 45.000
Produzione: Interstel
Distribuzione: C.T.O. (Via Piemonte 7/F, 40069 Zola Predosa - 051/753133)

Dopo il successo ottenuto sul numero precedente dal colossale *Universal Military Simulator* di Ezra Sidran, questo mese è il turno di *Empire*, quarto programma della serie *Starfleet* prodotta dalla software house americana Interstel.

Il programma tratta della lotta per la supremazia su uno dei pianeti la cui conquista era stata oggetto del secondo e terzo titolo della serie *Starfleet* (non importati in Italia). Per il dominio del pianeta Arabella possono competere sino a tre giocatori, e in mancanza di avversari umani può provvedere il computer a ricoprire i loro ruoli.

La caratteristica più importante di *Empire* sta nella sua semplicità d'impiego: anche se il gioco arriva a simulare nelle fasi più avanzate i movimenti di diverse centinaia di unità, il loro sistema di comando permette di controllarle facilmente anche senza aver letto il pur interessantissimo manuale (fornito, come con tutti i prodotti C.T.O., anche in versione italiana).

Il gioco è ambientato sulla superficie di un pianeta completo di mari, isole, fiumi, città, laghi e altri elementi geografici, che può essere progettato dallo stesso giocatore. Lo scopo ultimo di *Empire* è la conquista di un certo numero di città, che varia a seconda del livello di difficoltà prescelto.

Dopo aver selezionato in un'ap-

posita schermata il livello di abilità dei vari partecipanti, il gioco comincia. Nelle prime fasi i giocatori possiedono soltanto una città, ed è dai suoi cantieri che ottengono le armi e i mezzi con i quali potranno conquistare il pianeta. Ogni nucleo urbano produce divisioni corazzate terrestri, e sarà il computer a chiederci alla fine di ogni turno eventuali modifiche della produzione.

Le divisioni corazzate, che costituiscono il fulcro di *Empire*, sono i mezzi con i quali ogni giocatore esplora il terreno. Al passaggio dei carri armati, infatti,

re e conquistando eventuali città neutrali.

Il momento del combattimento fisico, per quanto differito, costituisce in ogni caso il nucleo di qualsiasi programma di guerra. *Empire* non fa eccezione, ma gli scontri armati vengono gestiti in maniera automatica, lasciando al giocatore soltanto il compito di muovere le proprie forze secondo la strategia preferita.

Dal momento che ogni città produce mezzi militari sino a che non viene fermata da un attacco nemico, in breve le forze in campo diventano tali e tante che è



la mappa di gioco rivela a poco a poco la sua struttura. Generalmente passano molti turni di gioco prima che si aprano le ostilità, e questo periodo è fondamentale per ottenere una certa conoscenza del mondo in cui ci si trova e per conquistare alcune città neutrali da convertire in centri di produzione.

Con semplici movimenti del mouse, è possibile ordinare alle città di produrre cacciabombardieri, navi di varia stazza e sottomarini, che andranno a rinforzare il nostro esercito, colonizzando contemporaneamente nuove ter-

facile perderle di vista. Per questo i programmatori della Interstel hanno fornito ogni pezzo sul campo di una finestra di comando, tramite la quale è possibile delegare al computer il controllo dell'unità secondo uno schema prefissato. I comandi comprendono funzioni di pattugliamento, ricognizione, trasporto, incolonnamento e molte altre, e una volta presa questa precauzione non dovremo più preoccuparci del pezzo sino a che non entrerà in combattimento.

Empire è un programma estremamente complesso che è stato

tuttavia realizzato in modo tale da risultare sempre chiaro, semplice da controllare e comprensibile in ogni sua parte. In un periodo in cui programmi come *Typhoon of Steel* riescono a complicare oltre misura le simulazioni belliche, costringendo il giocatore a gestire personalmente persino la ricarica dei proiettili, questa caratteristica ci sembra più che positiva.

A corredare il blocco principale del programma contribuiscono numerose opzioni. Per esempio, come negli altri giochi della serie *Starfleet*, è prevista la registrazione permanente della carriera militare di ogni giocatore, una trovata che si dimostra un simpatico incentivo alla competizione. La possibilità di ridisegnare la mappa del gioco e ridefinire alcuni suoi parametri garantisce inoltre una certa longevità al prodotto.

L'ultima opzione particolare, il cosiddetto "gioco per posta", non è invece altro che una delle più grosse ridicolaggini della storia dei videogiochi. Utilizzando questa "innovativa" possibilità, infatti, il giocatore dovrebbe salvare la situazione di gioco dopo ogni mossa, e spedire il disco con il file così ottenuto al giocatore successivo, che a sua volta farebbe la sua mossa e rispiederebbe il disco in un ciclo infinito. È noto che simili tecniche di gioco vengono da sempre usate per le partite di scacchi o per altri giochi di strategia, ma da un semplice calcolo è risultato che con i tempi delle poste italiane una partita fra un giocatore di Genova e uno di Milano potrebbe durare qualcosa come 23 anni, scioperi, vacanze e ritardi permettendo.

Nonostante questa triste considerazione postelegrafonica, *Empire* rimane un piacevolissimo war game che ha la sua forza nella semplicità e nella flessibilità, e che merita un posto di rilievo in questa categoria di programmi nonostante il suo aspetto audiovisivo piuttosto povero.

F.R.

ITALY '90 SOCCER



Forse i mondiali del '90 sarà meglio seguirli sullo schermo Rai piuttosto che su quello dell'Amiga, ma intanto che aspettiamo...

Computer: C-64/128/Amiga

Supporto: Cassetta/Disco

Prezzo: L. 18.000/25.000/39.000

Produzione: Simulmondo Italia (Viale Berti Pichat 26, 40127 Bologna - 051/251338)

Non siamo ancora a metà del 1989, ma i preparativi per i mondiali di calcio stanno già sconvolgendo il nostro Paese, ed era naturale che fosse proprio una software house italiana la prima a pubblicare un programma ispirato ai mondiali.

Vista la carenza cronica di esponenti italiani fra i produttori di videogiochi, avrete già intuito che l'onore è toccato alla Simulmondo, unico esempio di software house indipendente della nostra penisola. Il programma, opera di tre fratelli, presenta un completo campionato di calcio, che si sviluppa come una serie di partite in stile arcade secondo uno schema poco originale ma sicuramente funzionale.

Italy '90 Soccer sfrutta come forma di protezione l'inconsueto sistema del "dongle", un piccolo elemento hardware che va inserito nella seconda porta joystick del computer per permettere un caricamento corretto. Nonostante l'aspetto poco professionale causato da un trattamento a caldo di fattura chiaramente artigianale, la chiave di protezione svolge il suo compito senza problemi e ci esenta dagli ormai abituali sistemi d'identificazione basati sul manuale.

Il caricamento, anche se un po' lento, avviene senza problemi. Dopo un paio di schermate di

mediocre qualità, giungiamo all'unico (ma funzionale) menu del gioco, che permette di selezionare diversi parametri, fra i quali la durata della partita, il numero di giocatori e il grado di abilità del computer, oltre a un paio di opzioni che diventano chiare soltanto in un secondo tempo. Ci riferiamo alla possibilità di caricare da disco diverse "strategie" di gioco e alla scelta fra il sistema di controllo per principianti e quello per esperti.

La prima di queste opzioni non viene spiegata con chiarezza nemmeno dal manuale, ma non se ne sente comunque la mancanza. La seconda, invece, svolge un ruolo particolarmente importante e in un certo senso distingue questo ennesimo programma calcistico da tutti i prodotti similari che da anni continuano a invadere il mercato. Infatti, mentre il modo Novice permette soltanto un semplice controllo dei giocatori in campo, che possono essere spostati via joystick nelle otto direzioni e calciare il pallone nella direzione in cui sono rivolti, selezionando invece il box Professional vengono messi a nostra disposizione veri e propri prodigi di acrobazia che permettono tiri estremamente spettacolari.

Colpi di testa, scivolote, tiri a effetto e rovesciate sono le nuove possibilità aperte da questa opzione, per le quali va pagato tuttavia un caro prezzo in termini di semplicità di comando per non dire di giocabilità. Tutti questi tiri vengono infatti attivati da una sequenza di combinazioni di posizioni del joystick e del suo pulsante, e solo una grandissima pratica rende attuabili queste tecniche senza spiacevoli effetti collaterali quali falli o spaventose cilecche.

Tornando al gioco vero e proprio, una volta scelti tramite un comune menu RGB a barre i colori delle maglie delle squadre, viene finalmente il momento di entrare in campo. L'azione è vista

lateralmente, con il campo che scorre orizzontalmente sullo schermo mostrandoci il suo fianco più lungo e una delle tribune.

I giocatori entrano in campo dal fondo dello schermo fra il clamore (simulato, s'intende) della folla: a questo proposito andrebbe fatta notare la qualità medio-bassa di tutti gli effetti sonori del gioco, che stranamente peggiorano di partita in partita.

Il programma, dotato di una grafica semplice ma chiarissima che riproduce persino le ombre dei giocatori in campo, è unico nel suo genere nel ricostruire la

play", ma in realtà si ottiene soltanto una sorta di "indietro veloce" tipo videoregistratore dell'ultima azione eseguita.

Non c'è molto altro da dire sullo svolgimento della partita se non che vengono rispettate tutte le regole ufficiali, e che il pallone ha la stranissima abitudine di roteare anche da fermo, prima che l'azione abbia inizio. Una particolarità di *Italy '90 Soccer* è il bizzarro comportamento delle squadre manovrate dal computer, che spesso e volentieri non riescono a segnare se teniamo fermi i nostri giocatori, mentre si

PAC-LAND



L'ultima originalissima avventura di Pac-Man in una disastrosa versione per l'Amiga

Computer: C-64/128/Amiga

Versione: Amiga

Supporto: Cassetta/Disco

Prezzo: L. 15.000/25.000

Produzione: Grandslam Entertainment

Distribuzione: Leader (Via Mazzini 15, 21020 Casciago - 0332/212255)

Sono pochi i personaggi dei videogiochi che possono vantare un successo pari a quello di *Pac-Man*, il giallo e tondeggiate protagonista del gioco di labirinto più conosciuto del mondo, che ha avuto l'onore di apparire in cartoni animati, fumetti, pubblicità e persino in parecchi film (chi non ricorda la sua improvvisa ed esilarante apparizione sugli schermi radar di *L'aereo più pazzo del mondo?*).

Da quando l'instancabile mastro fece la sua comparsa in un coin op estremamente primitivo, a opera della Namco, lo abbiamo visto sotto molte diverse incarnazioni. Alcune, come *Pac-Man jr.*, *Ms. Pac-Man* e *Super Pac-Man*, non erano altro che semplici elaborazioni del concetto originario di labirinto bidimensionale, ma altri giochi hanno presentato il giallo eroe sotto aspetti molto diversi. *Professor Pac-Man* ha tentato d'insegnare la matematica e l'inglese in centinaia di scuole americane, *Baby Pac-Man* è stato il primo caso di simbiosi fra flipper e videogiochi, e il recente *Pacmania* ha finalmente mostrato nella gloria del 3D tutta l'allegria famigliola di Pac-Man. Ma il titolo più originale della saga del divoratore giallo, almeno nella versione che si può ammirare nelle sale giochi, è senz'altro questo *Pac-Land*.

Il programma, che vanta una conversione per il C-64 prossima alla perfezione, abbandona il te-



difficoltà nel controllo di palla di una partita reale: non ci è dato di sapere se questa difficoltà sia intenzionale o no, ma abbiamo apprezzato il fatto che è necessaria una certa pratica prima di poter affrontare degnamente una partita.

Durante l'incontro, il gioco ci offre solo due opzioni. Oltre alla pausa, infatti, la pressione del tasto "R" ci dà la possibilità di ottenere un "effetto moviola" indubbiamente ispirato al glorioso *MicroProse Soccer* per il C-64. Il manuale ne parla come di "re-

rivelano avversari irriducibili quando tentiamo di eluderli.

Italy '90 Soccer, come tutti i prodotti realizzati dalla Simulmondo, presenta una simpatica idea realizzata con molta poca cura dei particolari, il che abbassa notevolmente il valore del programma. Il manuale, multilingue ma infarcito di errori nei disegni che illustrano gli schemi di tiro, è un esempio evidente di questa tendenza, come lo è la confezione, sulla quale compaiono scritte in inglese nonostante la destinazione del prodotto. **F.R.**

ma del labirinto per mostrarci Pac-Man in una veste più vicina a quella ideata dalla Hanna & Barbera per la serie di cartoni animati. Pac-Man ha braccia e gambe, e persino un elegante cappello piumato che indossa per tutta la durata del gioco.

In *Pac-Land* non ci sono più pillole da ingoiare, ma Pac ha lo stesso i suoi problemi. Il gioco consiste nel portare a termine una serie di viaggi nella magica terra dei videogame, che naturalmente sono funestati dalla presenza ossessiva degli immancabili fantasmi colorati. Ogni livello è composto da tre sezioni a scorrimento orizzontale, divise in due fasi di andata e una di ritorno.

La storia comincia nei pressi della tondeggiante abitazione di Pac-Man, e vede il nostro eroe attraversare scenari cittadini, foreste, paesaggi montani e deserti, ogni volta con pericoli diversi. I numerosi ostacoli che s'incontrano durante il cammino prendono tutte le forme possibili, ma che si tratti di spinosi cactus, profonde piscine, piattaforme mobili o idranti impazziti, abbiamo solo una possibilità di affrontarli: vista la presenza di un bel paio di vispe gambette sul nostro personaggio, possiamo infatti utilizzarle per spiccare salti prodigiosi che si controllano in modo piuttosto vario con un attento uso dei comandi. Mentre una semplice pressione del tasto di fuoco fa compiere al nostro amico un salto normale, una rapida serie di tocchi sul joystick aumenta la lunghezza del balzo, che nelle mani di un giocatore esperto può protrarsi anche per più di una schermata.

Questa gamma di movimenti è utile per raccogliere i bonus offerti dagli scenari, ma soprattutto per evitare i fantasmi Inky, Blinky, Pinky, Clyde e la loro sorellina Sue. In *Pac-Land* i simpatici inseguitori usano ogni sorta di aiuto meccanico per ostacolare

i progressi del protagonista. Superato il quadro-bonus della chiesa e giunti in un modo o nell'altro al limite destro del livello, troviamo una misteriosa porta sospesa nell'aria. Attraversandola con un salto entriamo nello spettacolare mondo delle fate, la cui bellissima regina ci fa omaggio di un paio di brillanti stivali. Con queste calzature magiche possiamo camminare nell'aria, e la nuova possibilità si rivela molto utile durante i viaggi di ritorno, per i quali il tempo assegnato è particolarmente ridotto.

Un Pac-gioco senza pillole

hanno mantenuto tutte le caratteristiche del coin op originale. L'impressione generale è quella di un programma scritto in Basic, la cui complessità si scontra all'atto pratico con una lentezza di esecuzione esasperante.

Gli scenari (ottimamente realizzati) scorrono sullo schermo con la fluidità dei messaggi di sistema di un Pet, un ritmo incostante che oltre a fare innervosire provoca rapidamente una preoccupante sensazione di mal di mare. Gli sprite, piuttosto diversi da quelli offerti nella versione originale, sono animati poco o niente, e



energetiche rovinerebbe un'onorata tradizione, e infatti anche in questa variante troviamo ad aspettarci qua e là enormi pastiglie lampeggianti con le quali possiamo vendicarci degli ossessionanti fantasmi. Ingoiandone una, non solo diventiamo temporaneamente invulnerabili, ma possiamo persino divorare i nostri avversari!

Parlando di realizzazione, dobbiamo purtroppo dare un giudizio estremamente negativo della versione per l'Amiga di *Pac-Land*, anche se i programmatori

quando a questi difetti aggiungiamo una musica che è soltanto una pallida imitazione di quella che dovrebbe riprodurre, il risultato complessivo è sufficiente a far maledire il giorno dell'acquisto del gioco.

Pac-Land è stato realizzato onestamente, e questo va detto senza mezzi termini. La spilla e il poster inclusi nella confezione non risolvono le sorti di un prodotto di livello amatoriale che sconsigliamo vivamente, nonostante l'ottima qualità della sua controparte per il C-64. **F.R.**

DRAW 2000

Finalmente un CAD di alto livello per l'Amiga, con tutte le opzioni necessarie per un impiego professionale

Computer: Amiga

Supporto: Disco

Prezzo: L. 275.000

Produzione: Aegis

Disponibile presso: Lago (Via Napoleone 16, 22100 Como - 031/300174)



Il pacchetto *Draw 2000* della Aegis, una casa che ha già prodotto notevoli programmi per l'Amiga dedicati alla gestione dei suoni e delle immagini, si inserisce in un settore raramente preso in considerazione dalle software house che si dedicano alla macchina Commodore. Se infatti esistono numerosi programmi in grado di muovere oggetti sullo schermo oppure capaci di creare pagine stampate di buona qualità, non capita spesso di sentir parlare di disegno tecnico sull'Amiga. Secondo un vecchio luogo comune, questo settore (come quello dei database) sembra essere di pertinenza esclusiva del mondo MS-DOS.

In effetti la grafica a 4096 colori rende l'Amiga più adatto al disegno creativo che a quello tecnico. Si tratta inoltre di un campo, quello del CAD, che sicuramente interessa e coinvolge un numero minore di utenti e che in genere viene utilizzato più per lavoro che per passione o per hobby.

Ma nonostante tutto, alcuni pacchetti per il disegno tecnico a poco a poco hanno fatto la loro comparsa, cercando di utilizzare al meglio le risorse dell'Amiga. Ciò significa che, per esempio, la grafica viene sempre utilizzata nel modo interlaced a 640 x 512 pixel, e l'assenza di un dispositivo di input professionale come la tavoletta grafica viene in qualche modo superata grazie al mouse e all'interfaccia user-friendly. Uno

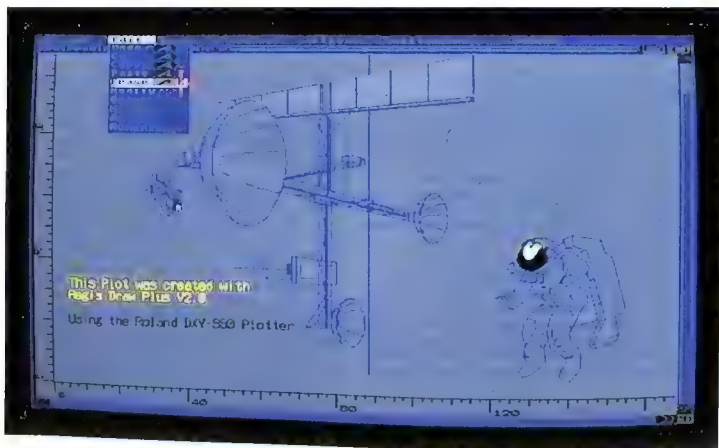
degli ultimi esponenti di questa nuova generazione di prodotti è *Draw 2000*.

Un programma di CAD permette di disegnare facendo uso di elementi precisi, in genere ben definiti geometricamente, ed è dunque adatto alla creazione di planimetrie, schemi elettronici e così via. Ovviamente, anche nel disegno creativo è possibile fare uso di elementi geometrici ma un rettangolo sarà sempre un insieme di pixel e non un'entità a sé stante. Come afferma nella sezione introduttiva il manuale di *Draw 2000*, la differenza tra i due

tipi di programmi sta nel fatto che un software pittorico non sa da quale istruzione è stato colorato un certo pixel, mentre un programma di CAD conosce esattamente la storia e le caratteristiche di tutti gli elementi entrati a far parte del disegno. È evidente che i due tipi di programmi si rivolgono a un mercato fondamentalmente diverso.

La differenza tra i due tipi di programmi sta nel fatto che un software pittorico non sa da quale istruzione è stato colorato un certo pixel, mentre un programma di CAD conosce esattamente la storia e le caratteristiche di tutti gli elementi entrati a far parte del disegno. È evidente che i due tipi di programmi si rivolgono a un mercato fondamentalmente diverso.

Draw 2000 viene mandato in esecuzione dal *Workbench*. Si può scegliere fra tre modi grafici; quello di default è in interlaced con la risoluzione di 640 x 512 pixel. Questo fa sì che la prima impres-



sione sia piuttosto negativa: il video sfarfalla vistosamente e questo non lascia presagire niente di buono. Per fortuna la qualità del programma mette in secondo piano gli inconvenienti dell'interlaced.

Il sottotitolo di *Draw 2000* lo definisce un "desktop CAD", e questo è in effetti quello che maggiormente colpisce quando lo si utilizza. Una delle caratteristiche che faceva preferire agli utenti i pacchetti di grafica pittorica piuttosto che quelli di CAD era sempre stata l'ostilità dei comandi da impartire per raggiungere i

Il sottotitolo di *Draw 2000* lo definisce un "desktop CAD", e questo è in effetti quello che maggiormente colpisce quando lo si utilizza. Una delle caratteristiche che faceva preferire agli utenti i pacchetti di grafica pittorica piuttosto che quelli di CAD era sempre stata l'ostilità dei comandi da impartire per raggiungere i

Il sottotitolo di *Draw 2000* lo definisce un "desktop CAD", e questo è in effetti quello che maggiormente colpisce quando lo si utilizza. Una delle caratteristiche che faceva preferire agli utenti i pacchetti di grafica pittorica piuttosto che quelli di CAD era sempre stata l'ostilità dei comandi da impartire per raggiungere i

risultati voluti: mentre per disegnare bastava premere il pulsante del mouse e "spruzzare" il colore sul video come con un pennarello o un aerografo, per ottenere un arco sul reticolo di *AutoCad* (il ben noto programma di disegno tecnico) occorreva impartire i comandi da tastiera dopo averne studiato la sintassi su un manuale in inglese. *Draw 2000*, invece, consente finalmente di creare disegni tecnici ricorrendo alla comodissima interfaccia utente dell'Amiga.

Il programma lavora su uno schermo proprio, sovrapposto a quello del *Workbench*, quando utilizza l'alta o l'altissima risoluzione, ma può anche lavorare direttamente sul *Workbench*. Utilizzando la massima risoluzione a

velocità di tracciamento (a dire il vero esiste un inconveniente anche in quest'ultima scelta: *Draw 2000* non sa di trovarsi su un *Workbench* in interlace e dunque non applica il giusto rapporto di scaling ai particolari grafici).

Una particolarità del funzionamento sullo schermo del *Workbench*, che dipende probabilmente da un problema di conversione da NTSC a PAL, è che su questo schermo le finestre di *Draw* possono arrivare a 256 pixel di altezza mentre negli altri non superano le 200 linee tipiche del sistema americano.

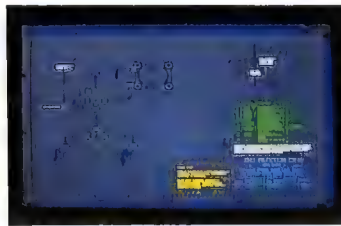
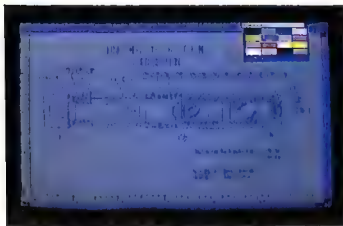
In tutti i modi grafici è presente nella parte superiore dello schermo una barra, che dà alcune informazioni sullo stato del sistema oppure cede il posto alle voci

l'opzione *New Drawing* del primo menu, *Project*, per avere a disposizione un foglio pulito con tutti i parametri impostati ai valori di default. O meglio, al valore specificato nel file *Draw.Defaults* (in formato ASCII) il quale, naturalmente, può essere modificato dall'utente.

Ora sul foglio sono presenti solo una griglia di riferimento e una coppia di rigelli sul lato inferiore e su quello sinistro, a simulare un tecnigrafo. Possiamo scegliere uno strumento di disegno dal menu *Tools* e cominciare a disegnare. Esistono due possibilità per tracciare i disegni: il mouse, e la pressione del tasto *Escape* dopo la scelta di un tool. In quest'ultimo caso appare una linea di comando vecchio stile sulla quale i parametri numerici si devono inserire da tastiera. Preferiamo comunque impiegare il mouse, che ci sembra molto più immediato.

Per prima cosa proviamo a tracciare una riga: il sistema conferma l'avvenuta scelta del tool *Lines* scrivendo questa parola in alto a sinistra. Dopodiché puntiamo il mouse in vari punti dello schermo e a ogni pressione del tasto sinistro compare un segmento di retta che parte dal punto precedente e arriva a quello attuale. Per abbandonare il tool basta premere due volte di seguito lo stesso pulsante, e le linee tracciate entreranno a far parte del disegno come oggetti.

Quasi subito abbiamo notato una stranezza: il movimento a passi discreti del puntatore. Per la precisione, la freccetta si muoveva soltanto da un'intersezione del reticolo all'altra senza fermarsi mai in punti intermedi. È in quel momento che ci siamo ricordati di quanto diceva il manuale a proposito di un'utile funzione chiamata *Grid Snap*, che in pratica serve per variare automaticamente le coordinate dell'ultimo punto raggiunto dal puntatore e



disposizione si lavora piuttosto bene e la quantità di pixel e di colori (16) a disposizione riesce a definire anche i particolari più minuti. Il problema dello sfarfallio si può attenuare con un monitor ad alta persistenza, ma comunque è possibile ridurlo notevolmente utilizzando particolari combinazioni di colori. Se si passa al modo 640 x 256, si elimina il problema dello sfarfallio e si risparmia memoria chip, però si comincia a lavorare male. Il modo *Workbench* è analogo a quest'ultimo ma fa uso di quattro soli colori, e quindi tutte le operazioni di tracciamento risultano più rapide; se poi si fa uso di un *Workbench* in interlace, si uniscono alta risoluzione ed elevata

del menu principale quando si preme il pulsante di destra. Ognuna di queste voci sottintende come al solito altre opzioni, e ciascuna di queste può essere raggiunta oltre che attraverso i menu e tramite combinazioni di tasti, anche per mezzo di un "Fast Menu". Un'altra caratteristica pregevole è la selezione contemporanea (tramite il pulsante di sinistra) di due o più opzioni, i cui effetti si manifestano in sequenza quando si rilascia il tasto destro del mouse.

Senza perdere tempo a parlarvi dei vari demo inclusi nel pacchetto, passiamo direttamente a esaminare le varie opzioni disponibili per costruire un disegno. A questo scopo, basta selezionare

farle coincidere con il nodo più prossimo della griglia, allo scopo di garantire una precisione che sarebbe irraggiungibile con il disegno a mano libera.

In un pacchetto di CAD la posizione dei pixel sullo schermo è solo un modo per rappresentare un insieme di elementi grafici che esistono nella memoria del computer come luoghi geometrici e insiemi di valori numerici. I pixel servono solo a dare una forma il più accurata possibile a questi oggetti, ma la stessa funzione può essere svolta da altri dispositivi, come il plotter. Può accadere che due punti che noi vediamo come adiacenti sul video siano in realtà distanti in modo apprezzabile a livello di coordinate: per esempio i lati di un quadrato non si toccano nei vertici. Lavorando con il Grid Snap attivo, la macchina ci assicura che i punti sono sempre quelli giusti. Se occorrono più punti "raggiungibili", si possono sempre diminuire le dimensioni delle maglie della rete.

Un secondo metodo di correzione automatica è offerto dall'opzione Data Snap, che rende "magnetici" i vertici di oggetti preesistenti sul disegno. Se con il mouse ci si avvicina oltre un certo limite al vertice di un oggetto già disegnato, il computer ne deduce che vogliamo toccare esattamente quel vertice.

Per la prova, comunque, abbiamo preferito eliminare entrambe queste possibilità. Tra l'altro, per sapere esattamente quali sono le coordinate del cursore, si può richiederne la visualizzazione in tempo reale sulla barra della finestra principale di *Draw 2000* (pur pagando lo scotto di un certo rallentamento). Inoltre si può attivare al posto della crocetta del puntatore una croce composta da due linee parallele agli assi cartesiani e lunghe quanto i lati dello schermo.

Abbandonato il tool Lines, curiosiamo tra gli altri strumenti di

disegno. Si possono disegnare circonferenze, rettangoli, ellissi, archi di circonferenza (ma non di ellissi) e poligoni. Il metodo usato per definire le dimensioni e la posizione della circonferenza che intendiamo tracciare è quello classico e consiste nell'assegnazione del centro e del raggio. Abbiamo purtroppo constatato una certa lentezza nella costruzione delle circonferenze e degli archi anche se il manuale afferma che le circonferenze sono in realtà poligoni di 72 lati. Senza coprocessore matematico il tempo impiegato è piuttosto alto.

Per quanto riguarda gli archi, prima si assegnano raggio e centro come per i cerchi, poi si sceglie la lunghezza dell'arco muovendo il mouse a partire dalla seconda estremità del raggio impostato. Le ellissi invece vengono definite specificando le coordinate del rettangolo a esse circoscritto.

Gli altri tool dell'omonimo menu consentono operazioni come la cancellazione, il disegno a mano libera di alcune parti e l'inserimento di testo in varie dimensioni. I caratteri sono tratti da una fonte stroke, ossia una fonte che utilizza linee anziché matrici di punti per costruire il carattere (il che comporta una migliore qualità nella operazione di rotazione, ingrandimento, rimpicciolimento). Inoltre i caratteri stroke sono particolarmente adatti alla riproduzione su plotter, almeno quanto le normali fonti a matrice di bit sono adatte alle stampanti grafiche.

Esistono poi alcune operazioni di edit degli oggetti, analoghe a quelle del menu Edit, che servono a copiare, spostare, ruotare o alterare le dimensioni di ciascun elemento grafico; la differenza sta nel fatto che dal menu Tools si agisce sempre sugli oggetti individualmente, mentre dal menu Edit si agisce sugli oggetti di un'area rettangolare prefissata.

Due peculiarità molto interessanti di questo pacchetto sono gli strati (Layers) e le parti (Parts). La prima caratteristica ci consente di definire il nostro disegno su più piani (fino a un massimo di 250) e poi di sovrapporli. È quindi possibile spezzare un disegno in diverse sezioni, lavorare su ognuna di esse singolarmente e poi riunire l'intero disegno, pur conservando la separazione fisica delle diverse sezioni.

Con il sintetico nome di "parti", invece, si vuole indicare l'importante caratteristica che permette di creare oggetti complessi (parti) mettendo insieme elementi fondamentali come cerchi e linee. Queste parti vengono viste dal sistema come entità uniche, qualunque sia l'operazione per cui vengono impiegate; possono poi essere archiviate su disco in una libreria. In qualsiasi momento è possibile slegare da questo vincolo gli oggetti che costituiscono l'insieme, per esempio per apporare delle modifiche. Il disco dimostrativo di *Draw 2000* propone tra le altre cose una libreria d'esempio che contiene simboli elettronici.

C'è poi la possibilità di effettuare ingrandimenti e riduzioni (Zoom) anche ripetuti in diverse aree del foglio.

È infine disponibile un tool per la quotatura del disegno, che permette d'inserire le quote (complete di frecce piene o vuote) e le linee di riscontro semplicemente definendo i due punti di cui va riportata la distanza.

Altri aspetti del programma comprendono la possibilità di riempire automaticamente tutte le aree chiuse che vengono tracciate, la definizione di linee di diversa tramatura e spessore, la scelta dei colori per tutti gli elementi del tavolo di lavoro, la possibilità d'interpolare linee spezzate arrotondandone i vertici (curve di Bezier) e la creazione di "hook" (uncini) su una linea retta.

Per quanto riguarda l'interfaccia col mondo esterno, *Draw 2000* è anzitutto in grado di memorizzare in diversi modi i propri disegni: come immagini bit-map, come insiemi di oggetti e coordinate e infine come insiemi d'istruzioni standard per ottenere da un plotter la tracciatura del disegno. In alternativa, si possono anche memorizzare su disco le singole parti del disegno e i parametri di sistema. Il pacchetto riconosce molti plotter, tra cui quelli della serie DXY della Roland.

Draw 2000 può lavorare anche con 512K di memoria, ma 1 MB è praticamente indispensabile per aprire più di una finestra grafica. In ogni caso, quando la memoria a disposizione diventa poca, requester o messaggi di alert ci avvisano dell'impossibilità di svolgere determinate fasi di lavoro. Un particolare che ci ha lasciato piuttosto perplessi è l'alta percentuale di crash del sistema che si verificano quando si prova a cambiare la modalità di visualizzazione, in particolare quando viene chiamato in causa il *Workbench*.

Possiamo comunque concludere che il pacchetto è completo sotto tutti i profili: cerca sempre di sfruttare al meglio le caratteristiche dell'Amiga, traendosi d'impaccio in maniera egregia quando si scontra con i limiti hardware della macchina.

Riteniamo che l'impiego professionale a cui dovrebbe essere destinato un pacchetto come questo sia senza dubbio possibile con *Draw 2000*, come senz'altro avrà potuto notare chiunque già abbia utilizzato programmi di CAD per il suo lavoro. Pochissimi sono i particolari che meriterebbero d'essere rivisti, e l'unico neo degno di nota è rappresentato da una certa lentezza nelle operazioni svolte in modo interlaccato. Con il coprocessore matematico, però, se non si arriva a prendere il volo poco ci manca!

D.G.

TV SPORTS FOOTBALL



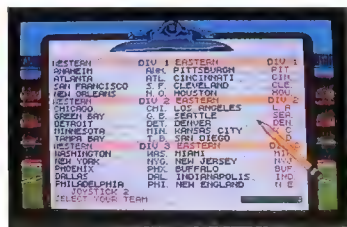
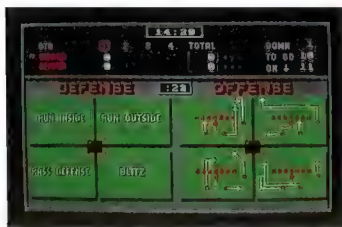
Finalmente un'ottima versione per home computer del football, il più spettacolare sport americano

Computer: Amiga
Supporto: Disco
Prezzo: L. 59.000
Produzione: Cinemaware (Mirror Soft)
Disponibile presso: Lago (Via Napoleona 16, 22100 Como - 031/300174)

Nato nel nordamerica verso la fine del secolo scorso come un ibrido a metà fra il calcio e il rugby, il football è uno degli sport più complessi e più violenti fra quelli che vengono praticati ai giorni nostri, ed è senza dubbio un ottimo rappresentante di tutto ciò che di solito viene identificato con l'America: spettacolo, colori, vivacità, dinamicità e so-

che di questa macchina permettono di realizzare un gioco che pur nella sua complessità e completezza, risulta veloce, facile da utilizzare, spettacolare e divertente.

Il programma, che s'inserisce nella fortunata serie Cinemaware, è dotato di caratteristiche grafiche eccellenti, quasi superiori agli standard a cui ci hanno abituati le sale giochi. La sigla d'introduzione, per esempio, è del tutto uguale alla sigla di una vera telecronaca televisiva, nella miglior tradizione Cinemaware. La finestra di gioco occupa la



prattutto soldi, tantissimi soldi.

La maggior parte dei tentativi di trasformare il football in un videogioco finora si sono scontrati con l'esigenza di dover riprodurre tutte queste caratteristiche, e hanno sempre dovuto sacrificare qualcosa per ottenere qualcosa: velocità, ricchezza grafica, complessità e varietà d'azione... scegliete pure, purché non vogliate tutto contemporaneamente.

Poi venne l'Amiga. *Tv Sports Football* testimonia in maniera inconfutabile che le caratteristi-

maggiore parte dello schermo, con una vista dall'alto che riprende tutta la larghezza del campo e circa 35 yarde di lunghezza, senza che la precisione della rappresentazione ne soffra.

Ma facciamo una sommaria carrellata su protagonisti e comprimari: al primo posto vengono i giocatori, che si muovono con la massima fluidità e sono stati realizzati molto bene anche dal punto di vista del disegno. Subito dopo viene l'arbitro, che compare soltanto in caso di falli e getta a terra il classico fazzoletto giallo

assegnando le relative yard e di punizione. A volte, poi, compaiono l'allenatore o altri personaggi, entusiasti o scontenti a seconda dell'andamento della partita, ma sempre in apprezzabili animazioni che costituiscono un simpatico momento di relax.

Quanto al gioco, l'azione è molto più rapida di quanto avviene di solito in programmi che riguardano questo sport tanto complesso, soprattutto grazie alla semplicità con cui si selezionano gli schemi di gioco. Al termine di ogni azione, infatti, compaiono sullo schermo quattro quadranti per la difesa e altrettanti per l'attacco che permettono, spostando il joystick in una direzione o nell'altra, di scegliere lo schema da utilizzare; effettuata la scelta, altri quattro quadranti consentono di scegliere tra diversi schieramenti della squadra nell'ambito dello schema prescelto. Oltre a queste 16 formazioni di base (quattro schemi per quattro schieramenti) si possono ottenere numerosissime altre varianti effettuando la selezione in modo leggermente diverso (per esempio tenendo un tasto premuto si ottiene uno schieramento speculare), ma sempre all'insegna della massima rapidità.

I modi di gioco dipendono dal numero dei giocatori (uno o due) e dal tipo di partita (partita di esibizione, partita di campionato, allenamento). Nelle partite vere e proprie vengono offerte quattro possibilità: esaminiamole una per una.

Soli contro il computer: in attacco si controlla il quarterback e il controllo passa poi al giocatore che riceve la palla, mentre in difesa si può scegliere di controllare uno qualsiasi dei defensive-back.

Due contro il computer: in questo caso un giocatore controlla il quarterback in attacco e un uomo a scelta in difesa mentre il secondo joystick controlla il linebacker

di sinistra in difesa e un wide receiver o un running back in attacco.

Due giocatori, uno contro l'altro: ognuno dei due controlla il proprio quarterback in attacco e un defensive back in difesa.

Il computer contro se stesso: questo modo è particolarmente utile per apprendere i rudimenti del football elettronico e i particolari stili di gioco.

In ogni caso, in qualsiasi dei sopraelencati modi si stia giocando, se un giocatore non prende iniziative entro 15 secondi, il computer si occupa di mandare avanti la partita da solo. Naturalmente basta impugnare di nuovo il joystick per riottenere il controllo della propria squadra. Questa caratteristica si rivela un enorme pregio di *Tv Sports Football* nei confronti di tutti i suoi predecessori, dal momento che evita le lunghe pause di riflessione che a volte si concedono i principianti, e dà il giusto rilievo alla continuità dell'azione. Una soluzione che si apprezza specialmente nelle partite a due, nelle quali spesso il tatticismo tende a prevalere sull'azione.

Se poi si è abbastanza modesti da ammettere che per imparare ci vuole del tempo, questa possibilità permette al giocatore di lasciarsi "guidare" dal computer nelle situazioni più difficili senza obbligarlo a subire le iniziative dell'avversario nei momenti d'incertezza. Teniamo a sottolineare il fatto che le scelte del computer sono sempre ispirate allo stile di gioco tenuto fino a quel momento dalla sua controparte umana, anche a costo di rinunciare a scelte che in quel momento potrebbero essere più favorevoli, non privando così la partita di quell'inconfondibile personalità che la rende qualcosa di diverso da una sterile sequenza di calcoli.

Un'opzione non particolarmente originale, ma comunque degna di nota, riguarda la possibi-

lità di formare una lega (l'equivalente di un campionato di calcio) a cui si possono "iscrivere" fino a 27 partecipanti, ognuno dei quali "allena" una squadra nella sua corsa verso l'ambitissimo superbowl.

Ma prima di poter aspirare al titolo di campione, sarà opportuno fare un po' di esperienza. A questo scopo è possibile giocare una partita amichevole fra due squadre, oppure fare pratica di azioni, passaggi o calci, selezionando l'opzione Practice all'inizio del gioco. Dopo essere diventati dei veri esperti, è il momento di confrontarsi con il gioco per vedere quali record riusciamo a battere.

Contribuiscono infatti ad arricchire lo scenario e a renderlo più realistico, cosa che renderà felici tutti coloro che amano questo sport, le statistiche che compaiono sullo schermo a metà tempo e alla fine della partita. I dati si possono richiamare anche durante il corso del gioco, in qualsiasi momento si vogliano riesaminare. Si possono ottenere dati estremamente dettagliati a proposito della squadra o del singolo giocatore, come vuole la tradizione americana: numero dei passaggi completi e incompleti, yard corse, percentuale di errore del quarterback e perfino graduatorie di merito relative ai singoli giocatori.

Insomma, se siete amanti del football, e siete sempre stati delusi dai videogiochi che vi hanno proposto, questo è il momento del riscatto! Se invece, pur non essendo degli esperti, questa recensione vi ha incuriosito, non lasciatevi spaventare da parole inconsuete come "quarterback" o "linebacker": *Tv Sports Football* rappresenta uno dei modi più semplici e divertenti per accostarsi per la prima volta a questo sport così complesso ma anche così spettacolare.

N.F.R.

R-TYPE E DENARIS



Il giallo di R-Type: tutto quello che avreste voluto sapere sullo shoot'em up più divertente del 1988

Computer: C-64/128/Amiga

Supporto: Cassetta/Disco

Prezzo: L. 18.000/25.000/49.000 (R-TYPE)

Prezzo: L. 15.000/25.000 (DENARIS)

Produzione: Electric Dreams (R-TYPE)

Produzione: Go! (DENARIS)

Distribuzione: Leader (Via Mazzini 15, 21020 Casciago - 0332/212255)

Quello che avete fra le mani è un numero innovativo di *Commodore Gazette*: da questo mese cercheremo di proporvi, quando è possibile, recensioni comparate dei programmi nelle due versioni per il C-64 e per l'Amiga.

Con questa recensione particolare, però, abbiamo pensato addirittura di riunire sotto un unico titolo ben due recensioni comparate, per un totale di quattro giochi analizzati contemporaneamente. La ragione di questa curiosa scelta sta nella tormentata storia che lega le vicende di *R-Type* e *Denaris*, e che ha costituito materia di scandalo già su altre pubblicazioni.

Tutto cominciò con la comparso nelle sale giochi di un innovativo coin op chiamato *R-Type*, verso la fine del 1987: il gioco suscitò immediatamente i consensi delle riviste specializzate e dei giocatori per l'originalità del suo schema e soprattutto della sua grafica (che diede origine alla definizione di "grafica organica"). *R-Type* fu immediatamente identificato dalle software house come un potenziale successo e dopo una breve ma intensa battaglia a suon di offerte fantasmagoriche, la casa inglese Electric Dreams riuscì ad acquistare i diritti di conversione per home computer del coin op.

La campagna pubblicitaria per promuovere *R-Type* cominciò nell'estate dell'88, con l'inserimento di piccoli poster nelle confezioni

di mezzo mondo. La realizzazione della conversione per C-64 venne affidata a un gruppo chiamato "Software Studios", e gli



Due schermate di *Denaris* nella versione per l'Amiga (sopra) e per il C-64 (sotto)

dei più importanti prodotti U.S. Gold (di cui la Electric Dreams è una divisione) e un afflusso continuo di fotografie e di comunicati stampa che invasero le redazioni

appassionati di videogame di tutto il mondo attesero con il fiato sospeso la pubblicazione del gioco, annunciata per Natale.

Durante questa lunghissima at-

tesa della buona novella, gli intelligenti ma truffaldini programmatori della Rainbow Arts pensarono bene di sfruttare la situazione. Nonostante avessero alle spalle una causa persa con la Nintendo per il loro *The Great Giana Sisters*, palesemente e malamente copiato dal famoso *Super Mario Bros* che ancor oggi costituisce il vanto della piccola console giapponese, il diabolico duo composto da Trenz ed Escher si mise a scrivere un clone di *R-Type*.

Il risultato degli intrighi binari dei programmatori tedeschi venne battezzato *Katakis*: il programma era di qualità eccellente, ma le similitudini grafiche e di schema con il coin op più famoso del 1987 fecero piombare sugli uomini della Rainbow Arts una doppia citazione in giudizio da parte della Electric Dreams e della U.S. Gold e una vera e propria pioggia di articoli negativi sulle riviste di settore di tutti i Paesi, che ostacolarono la distribuzione e la vendita del programma (realizzato sia per il C-64 sia per l'Amiga) sino a bloccarla.

La vicenda si sarebbe conclusa a questo punto, ma mentre le procedure legali sembravano essere giunte a un punto di stallo, la U.S. Gold si rese conto che i codici di *R-Type* in produzione nei loro studi non erano minimamente all'altezza di *Katakis*. Quello che accadde in seguito lo possiamo solo dedurre, benché non siano necessarie grandi capacità investigative: non potendo modificare in extremis i codici di *R-Type*, la Electric Dreams assunse gli autori di *Katakis* per riscrivere completamente la versione ufficiale del gioco, probabilmente in cambio del ritiro di ogni accusa e del controllo sulle vendite di *Katakis*. Comunque siano andate le cose, *R-Type* arrivò nei negozi firmato da Trenz & Escher.

Ora che tutti i potenziali acquirenti hanno rimpinguato le casse della U.S. Gold con i loro sudati

risparmi, la Go! (altra divisione della U.S. Gold) pubblica *Katakis*, dopo averne cambiato il nome in *Denaris*. Nel messaggio che appare dopo aver completato *R-Type*, gli autori affermano orgogliosi di averne scritto i codici in sei settimane. Quindi devono aver lavorato anche mentre era ufficialmente in corso il processo con la U.S. Gold, della cui soluzione legale nessuno ha mai saputo nulla. In questo momento Trenz ed Escher probabilmente se la stanno spassando con i proventi dei due (anzi, quattro) giochi, i

La trama vede un solitario caccia terrestre impegnato in una missione impossibile ambientata in un lontano futuro: i nemici da sconfiggere fanno parte dell'Impero Bydo, il cui mostruoso leader controlla personalmente tutti gli elementi della forza d'invasione grazie ai suoi poteri extrasensoriali. La viscida creatura è quindi il bersaglio da colpire per fermare istantaneamente e definitivamente l'invasione, ma non è possibile raggiungerlo se non passando attraverso una serie di difese biomeccaniche in cui crea-



Sopra: *R-Type* per l'Amiga. A destra: due schermate di *R-Type* in versione C-64

direttori del marketing della U.S. Gold stanno gongolando e i giocatori, benché vittime di una vera e propria truffa morale, hanno guadagnato due programmi eccezionali da aggiungere alla loro collezione. Vediamoli in dettaglio.

R-Type. Iniziando per correttezza dal programma originale, dobbiamo dire subito che gli sconvolgimenti legali relativi al gioco non hanno avuto che ripercussioni positive sulla sua realizzazione.

ture organiche e macchinari da battaglia sono fusi in entità simbiotiche tanto repellenti quanto letali.

Il veicolo a nostra disposizione è identificato dalla sigla R-9. Le sue caratteristiche principali sono l'ottima mobilità e le piccole dimensioni, che rendono più facile evitare i colpi avversari, e infine la capacità di agganciare strumenti modulari utili a migliorare le sue prestazioni. Gli apparecchi in questione possono essere agganciati sia anteriormente alla nave

sia posteriormente, e possono dotarla di missili a ricerca automatica del bersaglio, lanciafiamme, laser rimbalzanti, cannoni binari Balkan e una serie di altre armi. Grazie a una pressione della barra spaziatrice (o del pulsante del mouse sull'Amiga) è possibile separare con violenza il modulo delle armi addizionali, che viene così scagliato fra le forze nemiche seminando morte e distruzione con la sua corazza indistruttibile. Una semplice pressione dell'apposito comando è sufficiente a far rientrare il modulo, ma anche da sola la navicella non è del tutto disarmata: il veicolo R-9 è infatti dotato di laser standard e della possibilità di agganciare due scudi orbitali, e soprattutto dell'R-Type, l'arma che dà il nome al gioco.



In *R-Type* quello che affascina è senz'altro lo stranissimo aspetto grafico, che si unisce a una giocabilità perfetta per creare un gioco esaltante, difficile senza mai essere frustrante, e senza dubbio all'altezza del coin op originale. Se la versione per Commodore 64 può vantare delle colonne sonore identiche a quelle del coin op e una grafica con una definizione mai vista su questo computer, *R-Type* per l'Amiga utilizza al massimo le potenzialità grafiche della macchina con risultati a volte davvero stupefacenti. Sequenze come quella del gigantesco incrociatore spaziale che costituisce da solo un intero livello, o gli agguati degli scorpioni giganti nel cimitero alieno, sono tappe fondamentali nella storia



Dietro a questo misterioso nome in codice si nasconde un potentissimo cannone in grado di perforare con un solo colpo gran parte delle forze Bydo, ma che purtroppo ha alcune limitazioni. Per caricare l'R-Type, infatti, è necessario convogliarvi l'energia di tutte le armi, rimanendo così temporaneamente indifesi dagli attacchi nemici. L'uso del potentissimo cannone è tuttavia fondamentale per riuscire nella missione, in particolar modo quando viene il momento di affrontare i e disgustosi guardiani di fine livello: creature rettiloidi dotate di esoscheletro solo parziale, ammassi pulsanti di vene, occhi e tentacoli, o guardiani meccanici senza alcuna parte biologica.

della grafica dei videogame per l'Amiga, e hanno fatto premiare questo programma come la miglior conversione da coin op per l'Amiga di tutti i tempi.

Un'ultima nota di merito va alle musiche che accompagnano il caricamento di entrambe le versioni, di una qualità eccezionale specialmente per il Commodore 64.

Denaris. Se la nostra introduzione vi ha indotto a credere che si possa liquidare questo gioco come clone di *R-Type*, allora non ci siamo espressi con chiarezza. *Denaris* è un programma notevole, e le differenze tra le due versioni sono così marcate che si dovrebbe in realtà parlarne come di due programmi diversi, anche

se nella versione per il C-64 il gioco possiede uno scenario tanto ridotto da essere trascurabile e uno schema di gioco quasi identico a quello di *R-Type*. *Denaris* (o *Katakis*, come ancora molti lo chiamano) fornisce la possibilità di giocare in coppia, con un giocatore che prende il comando della grossa astronave principale e un altro che guida il modulo aggiuntivo: si tratta della principale differenza rispetto a *R-Type*, che permette schemi di gioco differenziati, impossibili nel gioco originale. Per quanto riguarda il sonoro, *Denaris* vanta le famose musiche semicampionate di Chris Heuelback, molto criticate ma indubbiamente innovative. Altri particolari: la definizione è inferiore a quella di *R-Type* e la grafica è in genere piuttosto grossolana, ma la sua velocità è assolutamente impressionante. Comunque la sua definizione si adatta particolarmente bene ai guardiani di fine livello, che mantenendo lo stesso spirito di *R-Type* rappresentano creature biomeccaniche grandi sino a mezzo schermo.

Denaris per l'Amiga (ma il programma "originale", *Katakis*, era simile a quello per il C-64) è tutto un altro gioco, diverso sia dalla sua versione per il Commodore 64 sia da *R-Type*. Rimane naturalmente la piccola astronave con il suo modulo aggiuntivo, rimangono le armi addizionali e gli scenari a scrolling orizzontale, ma il 90 per cento dei mostri biomeccanici è scomparso in favore di avversari più classici e la grafica ha un aspetto generalmente più monotono che nella versione "minore", per quanto la definizione sia ottima.

La sezione sonora non è nulla di eccezionale, e solo il fatto che non esistono altri prodotti simili (escluso naturalmente *R-Type*) le garantisce un certo successo. *R-Type*, in questo caso, è di gran lunga superiore.

F.R.

ZANY GOLF



Un bizzarro e divertentissimo minigolf elettronico costellato di laser, ventilatori e campi di forzer

Computer: Amiga

Supporto: Disco

Prezzo: L. 59.000

Produzione: Electronic Arts

Disponibile presso: C.T.O.

(Via Piemonte 7/F, 40069 Zola Predosa - 051/753133)

Sembra un periodo d'oro per i videogame ispirati al golf e alla sua più divertente e meno pretenziosa versione ridotta, il minigolf. Dopo le recenti esperienze interattive non troppo felici del nostro italico *Simulgolf*, la sempre grandiosa Electronic Arts propone ora la sua versione del minigolf: il golf goliardico.

Zany Golf è stato progettato e programmato da un gruppo di artisti indipendenti capitanato da un certo Will Harvey, un ex ricercatore dell'Università di Stanford affascinato dalle potenzialità grafiche dei computer: ossessionato dall'idea di produrre un programma semplice, divertente e spettacolare al tempo stesso, Harvey ha riunito i migliori artisti elettronici della West Coast, e il risultato è stato all'altezza delle premesse.

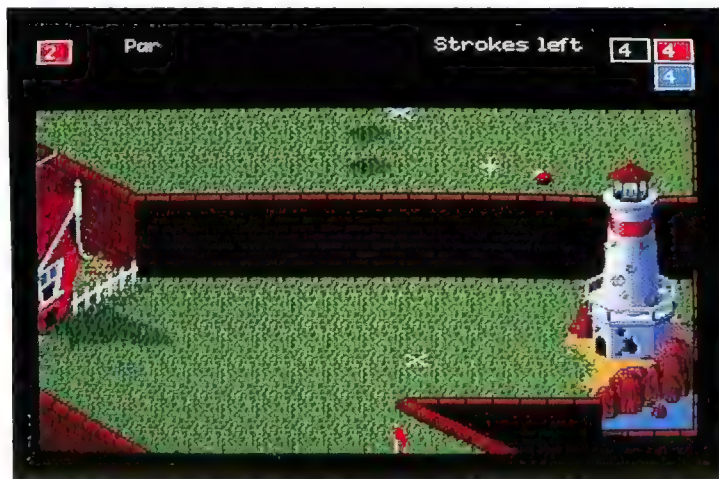
Zany Golf può essere giocato da un numero variabile di giocatori che può arrivare a un massimo di quattro. Ognuno parte con un numero fisso di tiri a disposizione, esauriti i quali viene eliminato dal gioco. Durante una partita è possibile conquistare colpi addizionali, che dipendono dall'abilità nel completare una buca.

Nella migliore tradizione dei giochi per l'Amiga, l'azione è interamente controllata dal mouse, e si divide su dieci percorsi che vanno affrontati sempre secondo

lo stesso ordine. Per ogni buca si può scegliere fra i due o tre punti di partenza proposti dal computer, e ogni percorso possiede un suo accompagnamento musicale che, per quanto breve, riesce a dare un'idea del tipo di percorso che ci aspetta e sfrutta molto bene le capacità sonore del computer. Vediamo ora quali ostacoli e quali trovate ci propongono le varie buche che compongono il percorso di *Zany Golf*...

Windmill (mulino a vento) è il nome della prima: il percorso è diviso in due parti ben distinte, e il nome deriva dalla presenza di

un vero e proprio emblema dello spirito del gioco: la palla deve seguire una traiettoria a U, e per far questo è necessario usare le capacità elastiche di... una gigantesca bottiglia di ketchup! A seconda della potenza del tiro, dal contenitore schizzano gocce di salsa in numero variabile che vengono convertite in tiri addizionali, e un'altra occasione di aumentare i colpi a nostra disposizione è data dalla possibilità di colpire un'innocente e minuscola fatina che svolazza sulla pista. Una volta arrivati alla buca, si scopre perché il percorso è stato



un mulino rotante che separa le due sezioni. Prima di affrontare ogni buca compare una schermata "introduttiva" in cui il computer ci dà alcune informazioni relative alla strategia da seguire. In questo caso il consiglio è di far entrare la pallina nella stretta porta del mulino, evitando le pale che la coprono periodicamente. Riuscendo in questa difficile impresa otterremo un certo numero di colpi-bonus, mandando inoltre la palla sulla traiettoria ottimale per raggiungere la buca.

Hamburger, la seconda buca, è

battezzato hamburger: la nostra meta è coperta da un colossale panino, oltretutto apparentemente indistruttibile. Per fortuna, ogni volta che premiamo il pulsante del mouse il panino si anima cominciando a rimbalzare. Dopo un certo numero di pressioni i balzi compiuti dall'hamburger sono tanto alti da lasciare libera la buca per un tempo sufficiente a prendere la mira, ma resta ancora un problema: ogni pressione sul pulsante del mouse corrisponde a un tiro, e non possiamo assolutamente per-

metterci di sprecarne troppi!

La buca successiva è una delle più semplici: Walls (muri), introdotta come sempre da una bella schermata statica, è ambientata su una ripida discesa divisa in tre fasce verticali. Partendo da uno stretto ripiano orizzontale posto nella parte più alta, dobbiamo far cadere la palla nella zona centrale, in fondo alla quale ci attende la buca. Le inevitabili complicazioni sono costituite questa volta da una serie di muri di mattoni che sbucano all'improvviso dal terreno deviando la pallina, ma un attento calcolo dei tempi permette di sfruttare la loro presenza a nostro vantaggio.

La buca successiva, chiamata Pinball (flipper) è divisa in due sezioni, e la prima è quella che ne giustifica il nome. La pallina viene lanciata come nel sempre più raro gioco da bar e viene fatta rimbalzare sul piano usando i pulsanti per controllare il movimento dei flipper veri e propri. Perdere la pallina equivale a un tiro sprecato, ma attivare la giusta sequenza di bersagli può fare aumentare notevolmente il numero di colpi a nostra disposizione. Lo scopo ultimo della sezione è comunque quello di far raggiungere alla pallina una buca dalla quale raggiunge poi la seconda parte del campo, un prato verde sul quale fanno bella mostra di sé i tre solidi che compongono il simbolo della Electronic Arts. Da qui la strada per la buca è semplice e diretta, e non ci sono problemi a raggiungere la buca successiva.

Fans (ventilatori) è un altro di quei percorsi in cui il mouse ha funzioni molto più ampie che non il semplice ruolo di sostituto della mazza. Il percorso si sviluppa su una ripida collina artificiale costellata di muretti e ventilatori: dopo il tiro ci viene richiesto di agitare il mouse per mettere in moto le pale degli apparecchi, che useremo per "soffiare" la palla

verso la buca, seguendo una traiettoria obbligata piena di salite e curve strettissime.

I tappeti magici della buca Carpets (tappeti) non hanno niente a che fare con quelli delle favole: le superfici a scacchi bianchi e rossi che costituiscono gran parte della collina su cui si trova la buca sono piuttosto degli strumenti elettromagnetici di grande potenza. La particolarità di questa buca sta nel fatto che quando la pallina giunge su queste superfici è possibile guidarla con ampi movimenti del mouse, facendole superare in modo poco ortodosso i numerosi ostacoli mobili che separano il punto di partenza da quello di arrivo.

La buca seguente si chiama Castle (castello), e il castello in questione è un vero e proprio castello da favola, inerpicato sulla sommità di una ripida distesa erbosa. Dobbiamo far entrare la pallina nello stretto portale d'ingresso della costruzione, ma il compito è reso particolarmente difficile dalla presenza di vaste distese d'acqua nelle quali è facile cadere perdendo il tiro e la pallina. Una volta entrati, tutto quel che ci separa dalla buca è un semplice percorso a zig zag che si supera giocando sui rimbalzi.

Anthill (formicaio) è l'ottava buca ufficiale del percorso, e si tratta di una collinetta sulla cui cima compare la bandierina che segna la posizione di una buca mobile. I movimenti della buca non sono preoccupanti quanto sembrano, e bisogna temere piuttosto le quattro discese a raggiera dalle quali la pallina può cadere una volta giunta in cima. Fortunatamente per noi, in fondo a ogni discesa troviamo un largo respingente idraulico: premendo il pulsante al momento giusto possiamo dare al nostro proiettile una spinta sufficiente a farlo tornare dove è venuto, ma per compiere questa manovra sono necessari riflessi notevoli.

Quella che dovrebbe essere l'ultima buca di *Zany Golf* è definita Energy, e per concludere in bellezza i progettisti si sono veramente sbizzarriti ottenendo un risultato grandioso. L'ambientazione è quella di un misterioso laboratorio da scienziato pazzo: al suono di una musica inquietante che ricorda i famosi motivi dell'ormai mitico *Marble Madness* ci troviamo circondati da campi di forza, piattaforme teletrasportanti, cannoni laser, computer, sfere di plasma e generatori di elettricità statica.

Purtroppo gli autori non danno alcun consiglio per aiutarci a superare la buca se non quello di colpire i grandi pulsanti del computer che costituisce il cuore del quadro... peccato che si trovino dietro a una barriera laser che polverizza la pallina in modo altamente spettacolare ma tragicamente definitivo. La buca, circondata da un gran numero di fori-trappola, si trova al piano superiore del laboratorio, dove si arriva solo attivando nella sequenza corretta una serie di teletrasportatori.

Avrete notato che abbiamo accennato a un'ipotetica decima buca, e infatti è possibile indovinarne la presenza anche dalla tabella dei punteggi, nella quale si trova una casella libera sotto alle nove corrispondenti ai percorsi che abbiamo descritto. Riuscire a raggiungerla è un'impresa non indifferente che richiede una lunga serie di manovre con i controlli della buca Energy, ma non sarà qui che scoprirete di che cosa si tratta.

Dopotutto, non è giusto togliere tutte le sorprese a un gioco tanto interessante. *Zany Golf*, che per la sua grafica, per l'eccellente sonoro e per l'altissima giocabilità può stare alla pari con un classico come *Marble Madness*, merita almeno questo atto d'omaggio da parte nostra.

F.R.

PURPLE SATURN DAY



Giochi olimpici spaziali con una bella grafica alla Moebius, ma senza grandi trovate

Computer: Amiga

Supporto: Disco

Prezzo: L. 40.000

Produzione: Infogrames

Distribuzione: C.T.O. (Via Piemonte 7/F, 40069 Zola Predosa - 051/753133)

Una volta all'anno, il pianeta Saturno si tinge di viola per uno strano fenomeno cosmico: questo bizzarro avvenimento ha diversi significati sui vari pianeti, ma per i saturniani e per i loro ospiti rappresenta il segnale d'inizio dei Giochi, l'equivalente cosmico delle Olimpiadi terrestri. In occasione del "giorno di Saturno viola", tutte le razze intelligenti della galassia mandano sul pianeta degli anelli i loro più grandi atleti perché affrontino le Quattro prove. E anche noi, che arriviamo dal vicino pianeta Terra, siamo qui per tentare di conquistare qualche alloro galattico.

Ogni atleta è contraddistinto da una serie di attributi (sullo stile dei giochi di ruolo) che influenzano le sue prestazioni. Agilità, Fluido mentale, Aggressività, Flusso temporale, Statura e Acuità visiva sono i fattori che fanno di ogni partecipante un elemento unico, e uniti al fattore di evoluzione possono essere visti come indicatori dell'abilità di ogni atleta.

Nella migliore tradizione dei giochi sportivi su computer, *Purple Saturn Day* offre al giocatore un menu principale dal quale è possibile esaminare i concorrenti e fare pratica nelle diverse gare prima di lanciarsi nella competizione. Le selezioni in questione vengono eseguite mediante un sistema di icone triangolari dall'a-

spetto leggermente alieno, che tradiscono l'identità degli autori.

Purple Saturn Day, infatti, è opera del gruppo Exxos, che ha contribuito - come molti ricorderanno - alla nascita dello stranissimo *Captain Blood*, del quale è stato mantenuto l'inconsueto stile grafico, paragonabile a quello di certi autori di fumetti francesi come Jean Giraud, meglio conosciuto sotto il nome di Moebius.

Le gare di *Purple Saturn Day* sono a eliminazione diretta, e chi raggiunge la vetta della piramide che appare fra una gara e l'altra conquista in premio un appassio-

La prima gara è quella delle "slitte elettroniche", con le quali dobbiamo dimostrare la nostra capacità di controllare l'energia cosmica. Il campo di gioco è un'arena rettangolare in orbita nello spazio, costellata di ostacoli cristallini che dobbiamo evitare con la nostra slitta. Al momento del via, viene lanciato un cristallo d'energia che dovremo colpire con i nostri cannoni laser. Ci lanciamo allora all'inseguimento, azionando il mouse o il joystick per pilotare lo strano veicolo su cui ci troviamo, e bersagliamo l'oggetto di colpi finché l'energia



natissimo bacio di miss Saturno, che sembra non avere molti problemi nell'abbracciare un Golgos ameboide piuttosto che il nostro prestante rappresentante umano.

Le gare possono essere affrontate in qualsiasi ordine, e tra un'edizione e l'altra dei giochi l'unica cosa che non cambia è la nostra razza umana e le prove da affrontare. Tutto sommato, l'impossibilità di ricoprire il ruolo di un alieno è deludente, ma non c'è tempo per lamentarsi della nostra mancanza di tentacoli: i giochi cominciano.

che contiene non si libera sul campo di gioco in una serie di frammenti. Sarà proprio il numero di questi frammenti che costituirà il nostro punteggio, una volta che li avremo raccolti.

Fra le regole non ben definite della gara non troviamo niente che ci impedisca di assalire il veicolo del nostro avversario, e una simile manovra ci permette anzi di sfondare la stiva della sua slitta e liberare un certo numero di cristalli, che possiamo aggiungere al nostro bottino. Una volta che tutti i frammenti sono stati

raccolti, un nuovo cristallo viene lanciato nell'arena sino a che il tempo previsto per il gioco non si esaurisce.

La gara delle slitte elettroniche ricorda moltissimo *Ballblazer* della Lucasfilm, nel quale si partecipava a una partita di calcio fra hovercraft spaziali. Nel caso attuale è stata aggiunta la possibilità di compiere curve di 90 gradi istantaneamente, ma questo non nasconde una monotonia di base che poteva essere giustificata a malapena per la versione a 8 bit di *Ballblazer*. Con una macchina come l'Amiga diventa inaccettabile.

La seconda gara è quella del "Salto in lungo nel tempo", per la quale conviene avere un attributo di Flusso temporale piuttosto alto. Lo scopo della gara è di catapultarsi con uno strano meccanismo alieno attraverso il tempo, dopo aver raggiunto una porta stellare che mette in comunicazione con i mondi di altre dimensioni.

La complessa meccanica del gioco è la seguente: una prima pressione del pulsante del mouse libera una serie di scintille nei pressi della nostra catapulta temporale. Muovendo un cursore sullo schermo, nuove pressioni del pulsante lanciano proiettili che devono intercettare le scintille. Dopo aver portato a termine quest'operazione per tre volte, otteniamo il controllo del motore della catapulta, rappresentato in una finestra come una molla simile a quelle che lanciano le palline nei flipper.

Comprimendo la molla e rilasciandola, il nostro veicolo viene proiettato verso una corona composta da sette soli, attraversata la quale ci si trova in una nuova dimensione dove si può soltanto continuare a far fuori scintille. E così via fino allo scadere del tempo a disposizione.

Lasciamo perdere le battute che si potrebbero fare su un

limite di tempo in una gara di salto nel tempo, e limitiamoci a notare che questa competizione è la più spettacolare del gioco, grazie alle sue sequenze di "salto" che sembrano tratte dai film di *Guerre stellari*.

La terza gara, il "Bowling cerebrale", ci vede impegnati nel controllo delle nostre capacità neurali: il campo di gioco è un gigantesco cervello biomeccanico diviso in due emisferi identici. Muovendo un cursore dal comportamento decisamente erratico, dobbiamo lanciare una serie di "cariche mentali" nell'emisfero di nostra competenza. Le cariche - che una volta raggiunto il loro bersaglio rimbalzano verso il cursore - hanno l'effetto di cambiare lo stato dei componenti elettronici che costituiscono gran parte del cervello. Aprendo e chiudendo interruttori e transistor con i nostri colpi di energia mentale, dovremmo riuscire a guidare le cariche elettriche che si muovono sulla superficie verso il condensatore che fa bella mostra di sé al centro del nostro emisfero. Una volta attivato questo componente possiamo assorbirne la forte carica elettrica, e attivare uno dei 18 chip attraverso i quali devono passare gli impulsi elettronici per raggiungere la loro meta finale: un ganglio centrale.

Anche in questa gara è possibile disturbare l'avversario lanciando cariche sul suo emisfero e rubandogli quelle contenute nel suo condensatore, nonostante sia molto difficile guidare correttamente i propri impulsi mentali.

Il gioco questa volta si svolge senza limiti di tempo, e diventa presto noiosissimo a causa della strategia usata dal computer, che tenta d'impedirci ogni progresso creando ossessive situazioni di stallo. Gli ottimi effetti sonori non riescono a risollevarne le sorti di un gioco basato su un'idea davvero eccezionale e demolito dalla tragica realizzazione.

Quarta e ultima gara dei Giochi è l'"Inseguimento sull'anello", che stabilisce quanto siamo abili a controllare lo spazio. Si tratta forse dello sport più "normale" fra quelli presenti ai Giochi, e funziona in realtà come una sorta di slalom parallelo. Il campo di gioco è quello degli anelli del pianeta, che sono composti da piccoli asteroidi. Oltre agli ostacoli naturali fanno la loro comparsa anche numerosi relitti spaziali, che sono poi l'elemento attorno al quale gravita il gioco.

I relitti, di colore rosso o giallo, giocano infatti il ruolo delle bandierine nello slalom, e si devono superare passando una volta sulla destra e una volta sulla sinistra. Alla guida del nostro veicolo, dal quale abbiamo una visione soggettiva e tridimensionale degli anelli, dobbiamo cercare di affrontare correttamente le porte senza schiantarci contro gli asteroidi, aiutati anche dall'indispensabile radar di bordo.

Ogni porta superata incrementa il nostro punteggio a condizione che la nostra navicella si trovi avvantaggiata rispetto a quella dell'avversario. In caso contrario è senz'altro più semplice e utile "tagliare" sul lato interno dell'anello, relativamente libero da asteroidi, dove potremo scatenare tutta la potenza del motore a fusione.

Purple Saturn Day ha un aspetto grafico molto ben curato (soprattutto nelle schermate statiche) e una presentazione quasi sempre di altissima qualità, ma come tutti i giochi sportivi o pseudo tali invecchia in fretta. Il numero ridotto di gare disponibili e l'impossibilità di affrontare avversari che non siano comandati dal computer, riduce molto il fascino alieno di questo programma.

Consigliabile soltanto agli amanti della bella grafica fine a se stessa.

F.R.

LAST DUEL



Ottima versione di un coin op a scorrimento verticale, da giocare anche in coppia

Computer: C-64/128/Amiga
Supporto: Cassetta/Disco
Prezzo: L. 15.000/15.000/25.000
Produzione: Capcom
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago - 0332/212255)

La Capcom, famosa azienda produttrice di alcuni dei migliori coin op degli ultimi tempi, è diventata una presenza molto più attiva anche nel mercato delle conversioni per home computer: nonostante i primi clamorosi fallimenti di alcune delle sue più attese conversioni (*Side Arms*, *Gunsmoke*), le capacità produttive della Go!, la sezione della U.S. Gold dedicata alla conversione di giochi Capcom, sembra essersi finalmente attestata su buoni livelli.

Dopo aver scritto alcune ottime routine grafiche per la versione domestica del semiconosciuto *L.E.D. Storm*, i programmatori della Go! hanno fatto un ottimo lavoro per questo "quasi-clone" di un coin op. Esaminiamolo in dettaglio.

La caratteristica principale della trama di *Last Duel* è... un errore. Per un'evidente svista del traduttore occupatosi della versione inglese delle schermate introduttive del gioco (originariamente in giapponese), la Stirpe di Galden è diventata la Tribù dorata (golden) del pianeta Bacula, colpevole dell'orrendo crimine di rapimento nei confronti della principessa Sheeta, regnante di Mu. Fortunatamente il nome preciso dei protagonisti non ha la minima importanza per lo svolgimento del gioco, e visto che della principessa non si vede nemmeno l'ombra anche portando a termi-

ne il gioco, conviene probabilmente limitarsi a esaminarne gli aspetti tecnici.

Last Duel è uno shoot'em up a scorrimento verticale, in cui uno o due giocatori attraversano una decina di livelli dalle più varie ambientazioni. I livelli ricadono in due categorie: i quadri dispari sono ambientati sulla solida superficie di vari planetoidi, mentre quelli pari si svolgono nelle nere profondità dello spazio. Questa distinzione ha una notevole importanza perché, come il recente

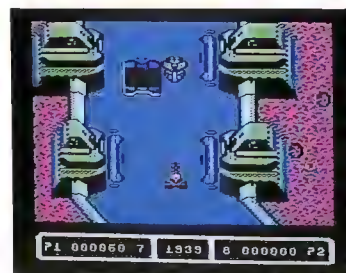
prima delle sue versioni è una futuristica automobile armata di lanciamissili, che può compiere balzi prodigiosi. Il veicolo è utile per superare le frequenti interruzioni del manto stradale, che è costellato di ampie botole nere e per di più spesso termina all'improvviso, cedendo il posto a un profondissimo baratro da superare per raggiungere la sezione successiva.

La seconda versione del veicolo è assolutamente identica a quella del secondo giocatore, e viene attivata automaticamente una volta giunti al termine dei livelli dispari.

Oltre ai pericoli costituiti dalle interruzioni della strada e dalle curve improvvise, ogni quadro scatena un vero e proprio arsenale contro i due giocatori: fra i nemici da eliminare troviamo automobili cariche di esplosivo, torrette mitragliatrici, carri armati, astronavi e molti altri avversari. Questi dividono accuratamente il loro campo d'azione a seconda che si trovino a terra o stiano volando, attaccando quindi il giocatore che si muove alla loro stessa altitudine.

Fortunatamente troviamo anche qualche aiuto: si tratta degli immancabili bonus, che si dividono anch'essi in due categorie. Mentre quelli indicati da una "T" hanno il trascurabile effetto di aumentare il tempo a nostra disposizione per raggiungere la fine del livello, che è comunque piuttosto elevato, le capsule marchiate "P" incrementano la potenza di fuoco del veicolo che le raccoglie, trasformando il singolo cannone iniziale in armi sempre più devastanti, per arrivare sino a un laser quinario (dotato di cinque direzioni di sparo contemporanee).

Una simile potenza d'attacco è quello che ci vuole per eliminare i colossali guardiani di fine livello, dotati di una gran varietà di forme e dimensioni. Fra loro i programmatori hanno incluso



La versione dedicata all'Amiga (sopra) si distingue solo per la migliore definizione

Silkworm, *Last Duel* offre alla potenziale coppia di giocatori due veicoli completamente diversi (e di conseguenza due stili di gioco che devono diventare complementari).

Se il secondo giocatore ha costantemente il controllo di una "normale" astronave dotata di laser, infatti, la persona che usa il joystick nella porta numero uno del computer si trova alla guida di un mezzo trasformabile. Nella

draghi a tre teste, gigantesche bestie scheletriche e mostruosi volatili biomeccanici, tutti dotati di una gran quantità di bocche da fuoco e quasi completamente invulnerabili.

I quadri ambientati nello spazio risentono delle ultime mode nella grafica degli shoot'em up più dei loro equivalenti terrestri: i giocatori si trovano spesso a dover affrontare creature vermiformi e viscidie barriere organiche, spesso in simbiosi con strutture metalliche gigantesche.

Last Duel è stato uno dei coin op più frenetici e spettacolari del 1988, e questa conversione mantiene (soprattutto nella versione per l'Amiga) entrambe le caratteristiche. La presenza di due veicoli completamente diversi l'uno dall'altro genera la necessità di elaborare schemi di gioco distinti a seconda del mezzo utilizzato, praticamente raddoppiando la longevità del gioco. L'azione frenetica non dà tregua se non durante le pause fra un livello e l'altro, e costringe i giocatori a un'attiva collaborazione durante l'attraversamento dei quadri di ambientazione spaziale.

In entrambe le versioni il commento sonoro si limita a una serie di effetti piuttosto banali di spari ed esplosioni. Un'eccezione curiosa è la musica che si sente con la schermata dei titoli, un vero e proprio "remix" della colonna sonora di *Thunderblade* della Ocean.

Nella versione per Commodore 64 mancano alcuni bonus rispetto alla versione a 16 bit, e lo scrolling è leggermente meno fluido che in *L.E.D. Storm* (che pure aveva una struttura molto simile a *Last Duel* e ne sfruttava parecchie routine), ma entrambe le versioni del programma brillano per velocità e varietà d'azione. L'acquisto è altamente consigliabile per tutti gli amanti dei giochi a gettone.

F.R.

SOFTWARE HELPLINE

Zak McKracken & the Alien Mindbenders: la soluzione

Se l'azione prevista per una data locazione va eseguita da un personaggio diverso da Zak, il nome di quel personaggio è indicato tra parentesi dopo il nome della locazione. Dopo il primo viaggio verso l'aeroporto e sull'aereo, la soluzione si limita a indicare il nome della destinazione successiva, omettendo le istruzioni per prendere l'autobus e usare il terminal delle prenotazioni. Ci sono diversi labirinti nella giungla che hanno struttura casuale e non possono essere mappati. In quei luoghi, limitatevi a continuare a svoltare fuori dal sentiero (anziché usare le uscite al limite sinistro o al limite destro dello schermo) tutte le volte che è possibile, sino a che non raggiungete la vostra destinazione. Alcune azioni devono essere portate a termine da personaggi particolari, come nel caso di Annie con la lettura della pergamena, ma la maggior parte possono essere eseguite da chiunque.

San Francisco: Camera da letto di Zak – Aprite il cassetto della scrivania. Prendete la bolletta del telefono (phone bill). Usate la lampada con il pesce (sushi in the fishbowl) senza accenderla. Tirate (pull) la tappezzeria strappata (torn wallpaper). Usate la tappezzeria sulla scheda in plastica (plastic card). Aprite l'altro cassetto. Prendete il cuscino.

Soggiorno e cucina di Zak – Prendete il cuscino sul divano (seat cushion). Prendete il telecomando (remote control). Aprite il ripostiglio (cabinet). Prendete la scatola di pastelli (box of crayons). Usate il pastello giallo (yellow crayon) sulla tappezzeria strappata. Aprite il frigorifero (refrigerator). Prendete l'uovo (egg).

Le Bakery – Premete il campanello (push doorbell) tre volte. Prendete il pane secco (stale bread).

Lou's Loans – Comprate la muta subacquea (wet suit), il set di attrezzi (tool kit), la mazza da golf (golf club), il cappello (hat), il naso finto (nose glasses) e la chitarra (guitar).

Bob E. Pins – Aprite la scatola degli attrezzi. Usate il tagliafilati (wire cutter) sul bobby pin sign.

Salotto di Zak – Usate la chiave serratubi (monkey wrench) sul tubo (pipe). Accendete (turn on) l'interruttore (switch) sul lavandino (sink). Usate il pane secco nel lavandino. Prendete le briciole (bread crumbs).

Compagnia telefonica – Indossate (put

on) il cappello e il naso finto prima di entrare. Entrate nell'ufficio. Aprite la porta del bancone (counter door). Usate il computer (computer terminal). Uscite dall'ufficio. Toglietevi (take off) il cappello e il naso finto.

Bus – Usate il kazoo. Usate la carta di credito (cashcard) nell'apposito lettore (cashcard reader).

Aeroporto – Date (give) la carta di credito all'incaricato (devotee) e prendete il libro. Usate il terminal delle prenotazioni (reservation terminal) per andare a Seattle.

Aeroplano – Andate in bagno e usate la carta igienica (toilet paper) nel lavandino (sink). Aprite l'acqua (turn on sink). Quando trabocca, premete il pulsante della hostess (call button). Andate nella parte anteriore (front) della sezione passeggeri. Prendete il cuscino del primo sedile (front seat). Prendete l'accendino (lighter). Se necessario ripetete il trucco della toilette per distrarre la hostess e aprite il forno a microonde (microwave oven). Mettete (use) l'uovo nel forno. Chiudete il forno. Accendete il forno (quando l'uovo esplose, la hostess rimarrà in quella zona). Aprite tutti i compartimenti (bin) sino a che non trovate una maschera a ossigeno (oxygen tank). Prendete la maschera.

Seattle: Parco nazionale – Prendete il ramo (tree branch). Date le noccioline (peanuts) allo scoiattolo (squirrel). Usate il ramo con la terra fresca (loose dirt).

Caverna – Usate il ramo sul nido (bird nest) [Per farlo avete bisogno di cercare alla cieca sia il nido sia le braci con il comando "What Is..."]. Usate il ramo sulle braci. Usate il nido sulle braci (fire pit). Usate l'accendino sulle braci e sul nido. Usate il pastello giallo sugli strani segni (strange markings). Andate sino alla porta con la croce egizia (ankh door). Salite sulla piattaforma e giratevi verso sinistra. Usate il telecomando. Prendete il cristallo azzurro (blue crystal). Volate a Miami.

Miami: Aeroporto – Date il libro all'ubriaco (bum) e prendete la bottiglia di whiskey. Usate il terminal per andare a San Francisco.

San Francisco: Cassetta postale – Usate il cristallo azzurro nella cassetta postale (drop slot) e attendete Annie.

Camera di Annie (Zak e Annie) – Date il whiskey ad Annie. Date la mazza da golf ad Annie. Date il tagliafilati ad Annie. Cambiate (switch) su Annie. Alzate il blocco (blotter). Prendete la carta di credito.

Bus e aeroporto (Annie) – Usate il tagliafilati sull'autobus. Usate la carta di credito nel lettore. Volate a Miami, poi al Cairo e infine a Kinshasa.

Kinshasa: Pista nella giungla – La mappa è casuale.

Capanna (Annie) – Date la mazza da golf

allo stregone (shaman) nella capanna all'estrema sinistra del villaggio. Osservate la danza rituale e scrivetevi l'ordine con cui i tre uomini si muovono, che cambia di partita in partita: si tratta del codice per aprire la Porta Enorme nella Stanza della Faccia su Marte. Volate al Cairo, poi a Londra.

Londra (Annie): Garitta – Date il whiskey

alla sentinella (sentry). Tirate l'interruttore (pull switch). Usate il tagliafili sulla recinzione (fence). Cambiate su Melissa.

Marte: Vicino allo Shuttle Bug, esterno (Melissa) – Aprite la porta ed entrate.

Interno dello Shuttle Bug – Prendete (pick up) lo stereo (boom box). Prendete il nastro (DAT). Aprite il portaguantini (glove compartment). Prendete il fusibile (fuse).

Prendete le carte di credito. Uscite.

Vicino allo Shuttle Bug, esterno (Melissa) – Date la carta di credito a Leslie.

Monolito (Melissa) – Usate la carta di credito sul monolito per due volte.

Entrata dell'ostello (Melissa & Leslie) – Usate il gettone (token) sulla piastra metallica (metal plate). Prendete il fusibile bruciato (burnt fuse). Usate il fusibile nella scatola dei fusibili (fuse box). Cambiate su Leslie. Andate da Melissa. Premete il pulsante di sinistra (che chiude la porta di sinistra). Premete il pulsante di destra (che apre la porta di destra). Entrate nella porta di destra.

Interno dell'ostello (Leslie) – Prendete il nastro di vinile (vinyl tape) nello scomparto di destra. Prendete la scala (ladder). Prendete le coperte (covers). Prendete l'alieno-scopa (broom alien). Aprite lo scomparto di destra. Prendete la torcia (flashlight).

Entrata dell'ostello (Leslie) – Premete il pulsante di destra. Premete il pulsante di sinistra.

Esterno dell'ostello (Leslie) – Usate l'alieno-scopa sul mucchio di sabbia (sand pile). Salite sull'autobus. Cambiate su Melissa. Salite sull'autobus.

Grande Faccia e Porta (Leslie & Melissa) – Usate la scala sulla porta. Premete i pulsanti sei volte, secondo la sequenza di Kinshasa. Prendete la scala. Entrate nella Grande Camera.

Grande Camera (Leslie & Melissa) – Entrate nella prima Porta Enorme.

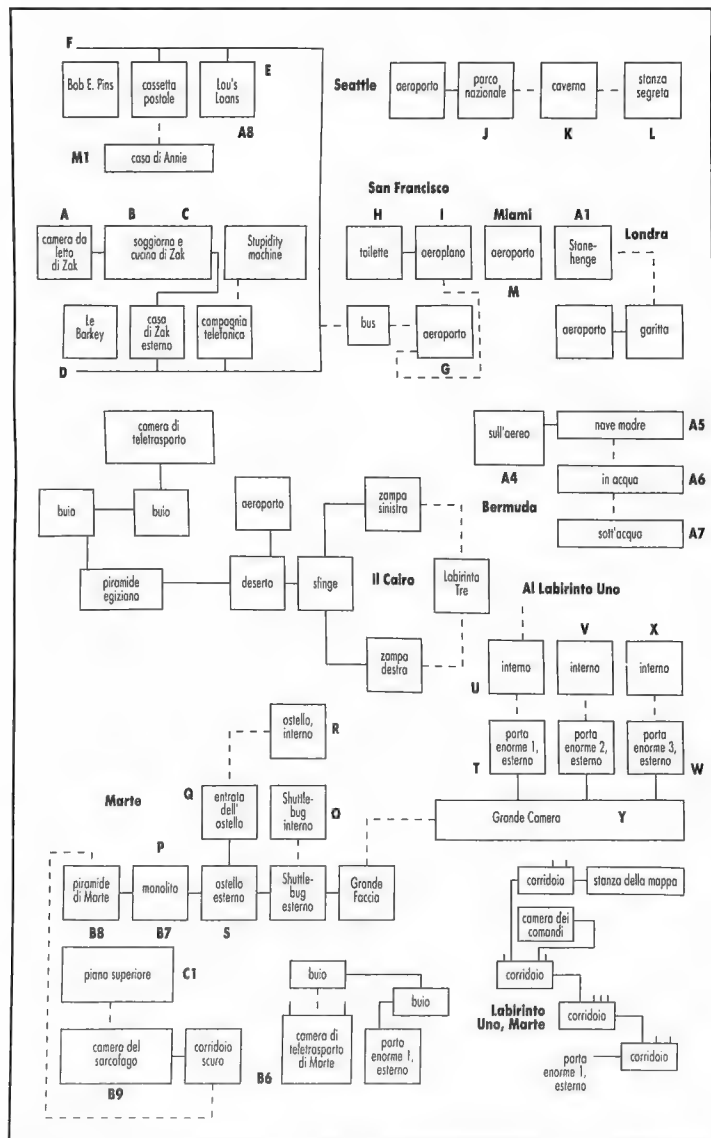
Porta Enorme Uno, esterno (Leslie & Melissa) – Date il nastro di vinile a Melissa. Cambiate su Melissa. Usate il nastro di vinile sul nastro DAT. Usate il nastro nello stereo. Accendete lo stereo (record).

Cambiate su Leslie. Usate la scala sul supporto (pedestal). Prendete la sfera di cristallo (crystal sphere). Prendete la scala. Accendete la torcia. Entrate nella porta verso nord.

Porta Enorme Uno, interno (Leslie) – Salvate il gioco. Seguite la mappa del Labirinto Uno sino alla Stanza dei Comandi. Accendete l'interruttore di destra. Accendete l'interruttore di sinistra. Toglietevi il casco (helmet) dopo che le condizioni si sono normalizzate. Seguite la mappa del Labirinto Uno sino alla Stanza della Mappa. Leggete gli strani segni [copiateli per usarli sulla sfinge in Egitto]. Seguite la mappa del Labirinto Uno per arrivare all'esterno della Porta Enorme Uno. Uscite con entrambi i personaggi nella Grande Camera. Entrate nella Porta Enorme Due.

Porta Enorme Due, esterno (Leslie & Melissa) – Date la torcia a Melissa. Cambiate su Melissa. Toglietevi il casco. accendete lo stereo (play).

Porta Enorme Due, interno (Melissa) – Prendete la croce egizia (ankh) oltre la porta all'estrema destra. Andate da Leslie. Date la croce egizia e la torcia a Leslie. Cambiate su Leslie. Uscite in direzione della Grande Camera. Entrate nella Porta



Enorme Tre.

Porta Enorme Tre, esterno (Leslie) – Usate la scala sul supporto. Prendete la sfera di cristallo oltre la porta all'estrema destra. Prendete la scala.

Porta Enorme Tre, interno (Leslie) – Usate la croce egizia sul pannello (panel). Prendete la chiave dorata (golden key).

Grande Camera (Leslie & Melissa) – Leggete gli strani segni sulla statua colossale fra le porte due e tre [copiate il simbolo per usarlo nel tempio messicano]. Mettete entrambi i personaggi vicini all'uscita della Grande Camera. Date la chiave dorata a Melissa. Cambiate su Zak.

San Francisco: (Andate dalla stanza di Annie all'aeroporto e volate in Messico).

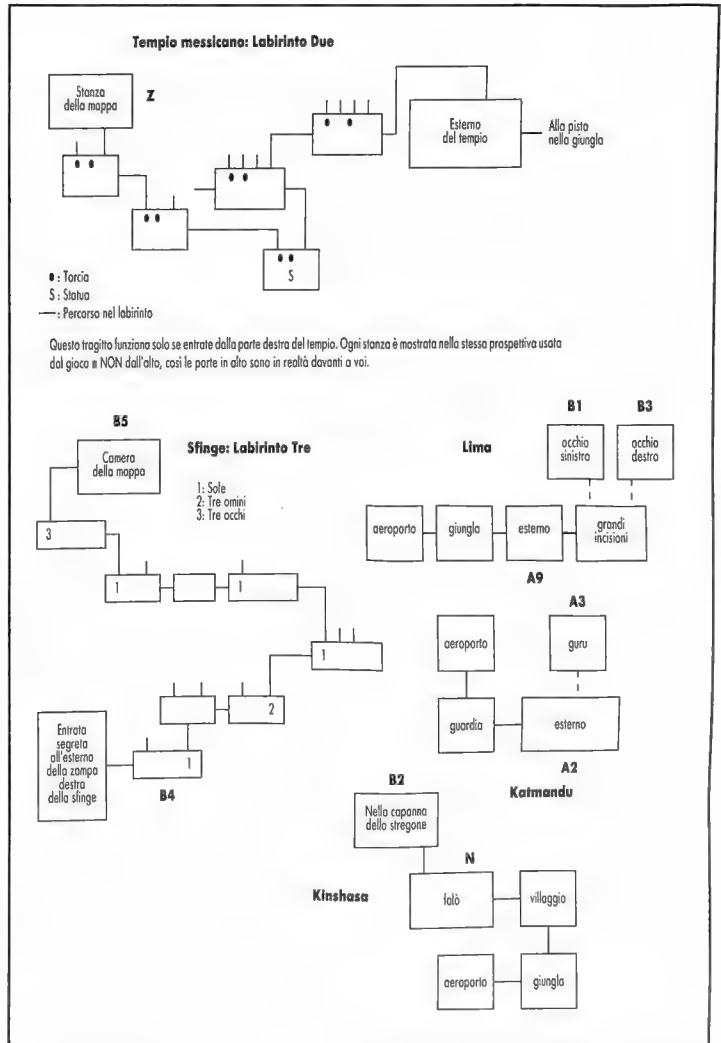
Mexico City: Sentiero nella giungla – Salvate. Questa è un'altra mappa casuale, ma ha diverse uscite sulle varie entrate del tempio. Continuate a provare sino a che non vi troviate sul lato destro del tempio, con la piramide sulla sinistra. Entrate nell'entrata del tempio sul lato destro dello schermo.

Nel tempio: Labirinto Due – Salvate. Usate il comando "What is..." per trovare le torce al buio, poi usate l'accendino sulle torce per illuminare ciascun corridoio. Seguite la mappa del Labirinto Due per arrivare nella camera della mappa. Prendete il frammento di cristallo giallo (yellow crystal shard). Usate il pastello giallo sugli strani segni e disegnate il simbolo trovato sulla statua di Marte [se necessario potete sempre cambiare su Leslie per rivederlo]. Seguite la mappa e uscite. Salvate e continuate a vagare sino a che non raggiungete l'aeroporto. Volate a Londra.

Londra: Stonehenge – Salvate. Usate il cristallo azzurro sull'altare [quando vi svegliate, muovete il cursore verso l'estrema destra dello schermo e premete rapidamente il pulsante due volte per andarsene prima che arrivi l'alieno]. Usate i frammenti di cristallo (tutti e due) sull'altare. Volate a Katmandu.

Katmandu: Usate l'accendino sulla paglia (hay). Prendete l'asta della bandiera (flagpole). Date il libro alla guardia. Andate dal guru per imparare come usare il cristallo azzurro. Usate il cristallo azzurro sullo yak dopo che il poliziotto è tornato alla prigione (jail). Cambiate su Zak e attendete che gli alieni vi gettino nella macchina della stupidità.

San Francisco: Compagnia telefonica – Mettetevi il cappello e il naso finto prima che gli alieni se ne vadano, in modo da evitare di perdere un sacco di tempo per far passare gli effetti della macchina. Aprite l'armadietto (cabinet) per recuperare i manufatti requisiti dagli alieni. Andate al piano superiore. Aprite il bancone. Uscite dall'ufficio. Togliete il



cappello e il naso finto.

Se non riuscite a eseguire le prime due azioni in tempo, fatelo dopo essere stati liberati dalla macchina ed essere stati scortati fuori. Quindi entrate negli uffici, prendete i manufatti e uscite. Potete usare entrambi i metodi tutte le volte che venite messi nella macchina della stupidità. Volate a Miami. Salvate. Volate alle Bermuda.

Bermuda: Aereo e hangar della nave madre – Aspettate sino a che non venite catturati dalla nave aliena. Premete il pulsante quando siete al suo interno.

Sulla nave madre col King – Leggete (read) il Lott-O-Dictor [scrivetevi il numero]. Date la chitarra al King [velocemente, subito dopo aver letto la macchina. Se riuscirete nell'impresa vi sarà mostrato come lasciare la nave e tornerete nella vostra stanza, altrimenti sarete gettati nella macchina della stupidità].

San Francisco: Lou's Loans – Comprate (buy) un biglietto della lotteria (lotto). Volate a Miami. Salvate. Volate alle Bermuda.

Bermuda: Sull'aereo – Aspettate sino a

che non venite catturati. Andate sul biplano finché è ancora nell'hangar. Usate il paracadute (parachute) quando cominciate a cadere.

In acqua (Zak & Delfino) – Usate il kazoo. Usate il cristallo azzurro sul delfino. Nuotate sott'acqua.

Sott'acqua (Delfino) – Prendete le alghe (seaweed) sulla destra. Prendete l'oggetto luminoso. Nuotate in superficie.

In acqua (Zak & Delfino) – Date l'oggetto luminoso (glowing object) a Zak. Dopo che siete stati catturati dagli alieni usate il cappello e il naso finto per recuperare gli oggetti che erano in vostro possesso).

San Francisco: Lou's Loan – Andate a controllare la vincita alla lotteria e prendete i 10 mila dollari di premio. Volate a Lima.

Lima: Pista nella giungla – Si tratta di uno dei labirinti casuali descritti nell'introduzione.

Altare ciba-uccelli – Usate le briciole di pane sull'altare che viene usato per cibare gli uccelli (bird feeder). Usate il cristallo azzurro sull'uccello.

Cielo (Uccello) – Volate sulle grandi incisioni. Volate sull'occhio sinistro. Prendete la pergamena (scroll). Volate nella valle. [Dopo che siete stati catturati, recuperate i vostri oggetti e tornate in questa locazione]. Usate il cristallo azzurro sull'uccello. Date la pergamena a Zak. [Verrete catturati nuovamente: dopo aver seguito la solita procedura volate a Londra].

Londra: Stonehenge (Zak & Annie) – Usate l'asta della bandiera sull'altare. Date la pergamena ad Annie. Cambiate su Annie. Leggete la pergamena. Cambiate su Zak. Prendete il cristallo giallo. Volate al Cairo e poi a Kinshasa.

Kinshasa: Sentiero nella giungla – Mapa casuale.

Capanna – Date il cristallo giallo allo stregone. Usate il cristallo giallo dopo aver puntato su Lima.

Lima: Occhio destro
Prendete il candelabro (candelabra). Usate il cristallo giallo puntando sul Cairo.

Cairo: Nella piramide egiziana, camera di teletrasporto – Usate l'oggetto luminoso sulla base. Usate il candelabro con l'oggetto luminoso. Tirate la leva (lever). Uscite dalla piramide [seguite la mappa] e andate verso la zampa della Sfinge (Sphinx leg) con gli strani segni.

Sfinge, esterno (Zak & Annie) – Usate il pastello giallo sugli strani segni [disegnate il simbolo trovato nella stanza della mappa su Marte]. Cambiate su Annie. Fate arrivare Annie da Zak, all'esterno della Sfinge. Entrate nella seconda porta (Annie).

Stanza della mappa della sfinge – Preme-

te il pulsante [secondo la sequenza che Annie ha appena letto]. Usate il pastello giallo sulla mappa fatta con la tappezzeria. Leggete gli strani segni [copiate i simboli trovati nella camera della Grande Faccia]. Usate il cristallo giallo puntando sulla Faccia, vi troverete nella camera di teletrasporto nella Grande Faccia di Marte.

Marte: Camera di teletrasporto – Usate il pastello giallo sugli strani segni [disegnate i simboli della stanza della mappa della sfinge]. Andate nella porta centrale (middle door). Muovetevi verso destra sino a che potete. Andate a sinistra per uscire [consultate la mappa del Labirinto Uno di Marte]. Avvicinatevi a Melissa e Leslie.

Grande Camera (Zak, Melissa & Leslie) – Cambiate su Leslie. Mettetevi il casco. Andate al monolito. Usate due volte la carta di credito sul monolito per ottenere due gettoni. Cambiate su Melissa. Mettetevi il casco. Andate al monolito. Usate la carta di credito sul monolito. Cambiate su Zak. Usate il nastro sigillatubi (duct tape) sulla bocca per i pesci. Mettetevi la bocca sigillata. Mettetevi la tuta subacquea. Mettetevi la maschera a ossigeno. Andate al monolito. Prendete un gettone.

Monolito (Zak, Melissa & Leslie) – Usate la carta di credito sul monolito (Zak). Mandate tutti i personaggi nello Shuttle Bug per rifornire le loro provviste di ossigeno [usate la valvola (oxygen valve)], poi tornate vicino al tram. Salvate. Usate il gettone sul tram [tutti e tre velocemente prima che il tram parta].

Piramide di Marte, interno (Zak, Melissa & Leslie) – Premete i piedi del sarcofago (Leslie). Salite le scale (Zak & Melissa), fate spostare Leslie dai piedi del sarcofago. Avvicinate Zak alla struttura di contenimento (containment device). Cambiate su Melissa. Usate la chiave dorata sulla scatola (box). Salvate. Premete il pulsante. Cambiate su Zak. Prendete il cristallo bianco [dovrete fare un po' di tentativi, perché avete poco tempo]. Usate il cristallo giallo puntando sull'Egitto. Toglietevi la bocca sigillata.

Piramide di Marte – Muovete Melissa sul lato sinistro delle scale. Premete i piedi del sarcofago (Leslie). Fate scendere Melissa e mettete entrambe le donne vicino al tram. Usate il gettone sul tram (Melissa & Leslie). Una delle due usi la carta di credito per ottenere un gettone. Usate il gettone per aprire la piastra metallica. Entrate. Prendete il fusibile. Fate entrare Melissa e Leslie nello Shuttle Bug. Usate il fusibile sul portaganti (glove compartment). Chiudete la porta. Usate i comandi (controls). [Se siete a corto di ossigeno, rifornitevi come spiegato prima. In realtà non importa che le ragazze lascino o meno Marte, ma la loro sorte cambia l'epilogo della vicenda]. Cambiate su Annie.

Cairo: Piramide egiziana, camera di teletrasporto (Zak & Annie) – Avvicinate Annie a Zak [attraverso il deserto e oltre

la porta della piramide]. Tirate la leva. Cambiate su Zak. Usate il candelabro con il cristallo azzurro. Usate il candelabro con il cristallo bianco. Usate il candelabro con il cristallo giallo. Accendete l'interruttore di destra. Cambiate su Annie. Accendete l'interruttore di sinistra.

LEGENDA DELLA MAPPA

A: Bolletta telefonica, boccia per pesci, lampada, mappa sulla tappezzeria, carta di credito, kazoo.

B: Cuscino, telecomando.

C: Pastello giallo, uovo, briciole di pane.

D: Pane.

E: Tuta subacquea, set di attrezzi (tagliafioli, sigillatubi, chiave serratubi), mazza da golf, cappello, naso finto, chitarra.

F: Insegna a forma di forcina.

G: Libro.

H: Carta igienica, lavandino.

I: Accendino, maschera a ossigeno.

J: Ramo, scoiattolo.

K: Nido, braci, strani segni.

L: Cristallo azzurro.

M: Frammento di cristallo giallo.

N: Sequenza per aprire la Porta Enorme nella camera della faccia di Marte.

O: Stereo, DAT, fusibile, carte di credito, valvola.

P: Gettoni.

Q: Fusibile bruciato.

R: Nastro di vinile, scala, alieno-scopa, torcia.

S: Mucchio di sabbia.

T: Supporto e sfera di cristallo.

U: Strani segni (copiare i simboli per usarli sulla sfinge in Egitto).

V: Croce egizia.

W: Supporto e sfera di cristallo.

X: Chiave dorata.

Y: Strani segni (copiare i simboli per usarli nel tempio in Messico).

Z: Frammento di cristallo giallo, strani segni.

A1: Altare.

A2: Paglia, asta di bandiera, yak.

A3: Guru (vi insegna a usare il cristallo azzurro).

A4: Paracadute.

A5: Numero vincente della lotteria.

A6: Delfino.

A7: Alghe, oggetto luminoso.

A8: 10 mila dollari.

A9: Portacibo e uccello.

B1: Pergamena.

B2: Stregone (vi insegna a usare il cristallo giallo).

B3: Candelabro.

B4: Strani segni.

B5: Geroglifici, strani segni (da usare nella stanza della Grande Faccia).

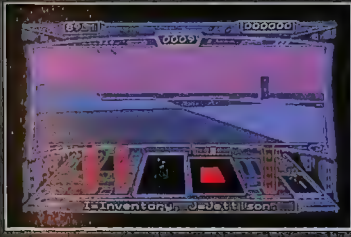
B6: Strani segni (disegnate i simboli dalla camera della mappa della sfinge).

B7: Tram.

B8: Serratura.

B9: Sarcofago e piedi del sarcofago.

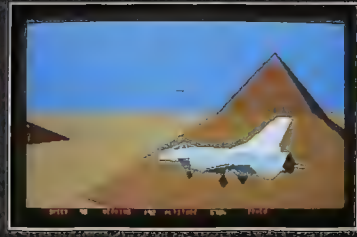
C1: Struttura di contenimento, scatola, cristallo bianco. ■



I PRIMI 10

programmi scelti dai lettori

- | | | |
|-----|---------------------|---------|
| 1. | Starglider 2 | (Amiga) |
| 2. | Pool of Radiance | (C-64) |
| 3. | R-Type | (Amiga) |
| 4. | R-Type | (C-64) |
| 5. | Dragon's Lair | (Amiga) |
| 6. | TrianGO | (C-64) |
| 7. | MicroProse Soccer | (C-64) |
| 8. | War in Middle Earth | (Amiga) |
| 9. | Rocket Ranger | (C-64) |
| 10. | Times of Lore | (C-64) |



IL PROGRAMMA DEL MESE

scelto dalla redazione

Falcon (Amiga)

[...] Tutti gli aspiranti piloti da caccia, che non vedono l'ora di sedersi davanti al proprio Amiga e volare, hanno la possibilità d'inserire nel proprio hangar un nuovo straordinario velivolo: l'F16 Falcon.

(da Commodore Gazette numero 1789)

MODALITÀ PER LA VOTAZIONE

Classifica Software, è un'occasione che Commodore Gazette dà ai suoi lettori per esprimere il proprio parere sui migliori programmi del momento. Per votare, compilare (anche solo parzialmente) il tagliando riportato in fondo alla pagina, ritagliare, e spedire a:

**Commodore Gazette, Classifica software
Via Monte Napoleone, 9 - 20121 Milano**

NOME _____
COGNOME _____
INDIRIZZO _____

SECONDO ME, I MIGLIORI PROGRAMMI DEL MOMENTO SONO:

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____

9. _____
10. _____

COMPUTER POSSEDUTO (marca e modello)

LEGGETE COMMODORE GAZETTE:

- Abitualmente
 Raramente
 Questa è la prima volta

INPUT / OUTPUT

INPUT/OUTPUT SVELA I SEGRETI DEL VOSTRO COMPUTER



C-64/128 in modo 64

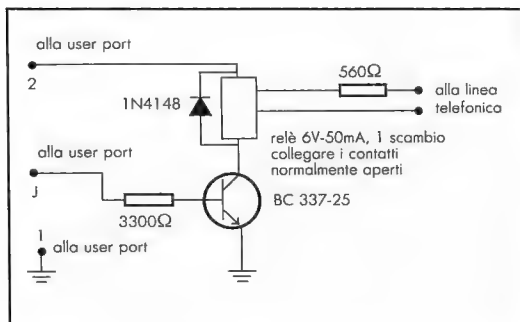
***193 Combinatore telefonico** - Questo progetto può essere un valido spunto per la creazione di una segreteria telefonica gestita dal vecchio C-64. Consiste di un circuito tramite il quale si può pilotare (grazie alla user port) un relè che provvede a generare impulsi telefonici, cioè a comporre numeri. Si veda la figura nella pagina seguente, che descrive lo schema hardware del circuito. Il software di gestione provvede a generare le giuste temporizzazioni per la parte elettronica.

```
10 REM Inizializza la user port
20 POKE 56579,32 : POKE 56577,0
30 REM Compone il numero 02/000000
40 A$="02/000000" : GOSUB 1000
50 REM A questo punto la linea e' attiva
60 WAIT 197,64,64
70 POKE 56577,PEEK(56577) AND 223
80 END
1000 REM Attiva la linea
1010 POKE 56577,PEEK(56577) OR 32
1020 FOR TT=1 TO 3000 : NEXT
1030 REM Genera gli impulsi per il numero in A$
1040 FOR TE=1 TO LEN(A$)
1050 T2=ASC(MID$(A$,TE)) - 48
1060 IF T2<0 OR T2>9 THEN I140
1070 IF T2=0 THEN T2=10
1080 FOR TI=1 TO T2
1090 FOR T1=1 TO 35 : NEXT
1100 POKE 56577,PEEK(56577) AND 223
1110 FOR T1=1 TO 25 : NEXT
1120 POKE 56577,PEEK(56577) OR 32
1130 NEXT
1140 FOR T1=1 TO 460 : NEXT
1150 NEXT
1160 RETURN
```

Prima viene inizializzata la user port, configurando la sua porta a 8 bit in maniera da disporre di una sola uscita (il bit definito "non assegnato" nella *Guida di riferimento per il C-64*) e lasciando inutilizzati gli altri sette (per garantire la compatibilità con un modem). Successivamente viene eseguita la subroutine alla riga 1000, passandole come parametro il numero telefonico da comporre: si noti che sono riconosciuti tutti i caratteri tra 0 e 9, mentre gli altri caratteri equivalgono a una pausa nella generazione degli impulsi. Questa routine rende attiva la linea telefonica chiudendo i contatti del relè, poi attende che la linea sia stabile e solo allora provvede a generare gli impulsi relativi alle cifre del numero che si desidera comporre. Alla fine la linea è lasciata attiva, per consentire la comunicazione. A questo punto l'utente può sollevare la cornetta del telefono e iniziare la conversazione, che verrà interrotta premendo un tasto qualsiasi.

Input/output rivela ogni mese ai suoi lettori routine di programmazione per C-64, C-128 e Amiga. Avete in mente un'idea particolare, una routine, un utile stratagemma di programmazione, avete sviluppato un breve programma o in generale pensate che le Vostre conoscenze possano interessare l'utenza Commodore? Allora scrivete, e inviate gli eventuali listati stampati su carta bianca o salvati su disco a:

COMMODORE GAZETTE
Input/output
Via Monte Napoleone,9
20121 - Milano



Commodore 128

***194 Scambio di variabili** - Siamo pronti a scommettere che sottoponendo a esperti programmatori il quesito "come fareste a scambiare tra loro i valori di due variabili?", otterremmo sempre la stessa risposta. Ossia, si prende una variabile temporanea, le si assegna il valore della prima variabile, si memorizza nella prima variabile il valore della seconda, e infine si imposta la seconda variabile al valore di quella temporanea; tutto questo è valido in qualsiasi linguaggio, con qualsiasi tipo di variabili. Esiste però un altro metodo, altrettanto universale, ed è il seguente:

```
10 A = 5 : B = 13
20 PRINT A,B
30 A = XOR(A,B)
40 B = XOR(A,B)
50 A = XOR(A,B)
60 PRINT A,B
```

Si noti l'uso dell'operatore XOR del Basic 7.0, che in tale linguaggio è camuffato da funzione ma raggiunge i medesimi risultati dell'istruzione EOR in linguaggio macchina, ossia restituisce l'OR-esclusivo dei due argomenti. L'OR-esclusivo è una funzione booleana che agisce sui singoli bit dei due operandi, restituendo un 1 quando il valore dei due bit corrispondenti è diverso, e uno 0 quando i valori sono uguali.

Il metodo descritto consente di evitare l'uso della variabile temporanea (anche se il tempo d'esecuzione aumenta) e funziona in qualsiasi linguaggio dotato dell'operatore XOR. Il tipo di dati che possono essere trattati dipende dal tipo di linguaggio, ma in generale quando è disponibile la funzione XOR a livello elementare, ossia di byte, è possibile scrivere opportune routine per estenderne l'uso a variabili di dimensioni maggiori come le variabili in virgola mobile e le stringhe.

SoftMail

VENDITA PER CORRISPONDENZA DI PROGRAMMI ORIGINALI PER TUTTI I TIPI DI COMPUTERS
 VIA NAPOLEONA 16 - 22100 COMO - TEL. (031) 30.01.74
 © SoftMail è un marchio registrato da Lago inc



ACCESSORI		
Copriastiera A500	25.000	Phantom fighter 45.000
Final cartridge III	110.000	Police quest 49.000
Flicker master	29.000	Populous telef.
Joy. Tac 5	30.000	Question II 56.000
MouseMat tappetino	22.500	R-type 49.000
Coprimouse	20.000	Rockat ranger 59.000
Portamouse	12.500	SEUCK telef.
Portadischi 3" (30)	34.000	Space harrier 29.000
Portadischi 5" (40)	37.000	Super hang-on 49.000
SlimLine (tastiera64)	49.000	Sword of sodan 69.000
		Triad Vol. 1 59.000
		Tiger road 25.000
		Trackcut manager 49.000
		TV sport football 59.000
		Univ. Military simul. 45.000
		War in middle heart 49.000
		Winter edition 25.000
		Workbench 1.3 45.000
		Zak McCracken 59.000
		Zany golf 59.000
		Zoetrope 199.000



Presentiamo in questa pagina alcune tra le ultime novità del catalogo SoftMail. Ecco qualche informazione utile sul nostro servizio: è possibile effettuare ordini subito un cliente molto speciale! SoftMail ti riserva informazioni, consigli, anticipazioni e tutta la gentilezza che ti meriti.

LIBRARIANTS & TIPS		
Bard's tale I	22.500	
Bard's tale II	25.000	
Bard's tale III	25.000	
Deathlord	19.000	
Dungeon master	25.000	
Elite	18.000	
Mars saga	19.000	
Pool of radiance	20.000	
Quest for clues	39.000	
Sierra on line: vari		
Ultima V	22.500	
Wasteland	16.500	

AMIGA		
Aegis draw 2000	275.000	
Audiomaster II	105.000	
Bal. of power 1990	49.000	
Barbarian II	39.000	
Blastaroids	29.000	
Cosmic pirate (ITA)	39.000	
Dakce photolab	250.000	
Designsaurus	69.000	
Digiview gold PAL	330.000	
Disk drive esterno	299.000	
Dragon's lair (1 Mb)	75.000	
Driller ITALIANO	59.000	
Dungeormaster I	59.000	
Elite	45.000	
Kennedy approach	59.000	
K. Daglish soccer b.	49.000	
Kristal	59.000	
LightCamAction!	99.000	
Modeler 3D	129.000	
Neuromancer	telef.	
Photon paint 2.0	telef.	
Photon video:		
Cell animator	199.000	
Transp. controller	490.000	

COMMODORE 64/128 DISCO		
Accolade tutti gli sports tel.		
Adv. dungeons&dragons tel.		
Barbarian II	25.000	
Battles of Napoleon	59.000	
Defender ... Crown	15.000	
Demaris	15.000	
Driller	29.000	
Dynamic duo	20.000	
Exploding fist+	18.000	
F14 Tomcat	29.000	
Fish	29.000	
G.Lineker hotshot	15.000	
Gran prix circuit	29.000	
Last duel	15.000	
Legend blacksilver	18.000	
Mars saga	35.000	
McArthur's war	49.000	
Microprobe soccer	49.000	
Neuromancer	39.000	
Night rider	21.000	
Renegade III	18.000	
Rockat ranger	39.000	
R-type	25.000	
SEUCK ITA	35.000	
Total eclipse ITA	20.000	
Typhoon of steel	59.000	
Ultima V	49.000	
War in middle heart	25.000	
Wasteland	39.000	
Who fr. Roger Rabbat telef.		
WWF superstars 1 e 2 telef.		
Zak McCracken	39.000	

ORA APERTI AL PUBBLICO

Buono d'ordine di inviare a: LAGO DIVISIONE SOFTMAIL, VIA NAPOLEONA 16, 22100 COMO, TEL. (031) 30.01.74, FAX (031) 30.02.14

Desidero ricevere i seguenti articoli:

Titolo del programma	Computer	Prezzo

G Spese di spedizione Lit. 5.000
 ORDINE MINIMO LIT. 25.000 (SPESA ESCLUSA) TOTALE LIT.

Pagherò al postino in contrassegno
 Addebitate l'importo sulla mia: CartaSI Mastercard Visa American Express
 Numero scad.

Cognome e nome _____ Nr. _____
 Indirizzo _____
 CAP _____ Città _____ Prov. _____ Tel. _____

FIRMA (Se minorenni quella di un genitore)
 Verranno evasi SOLO gli ordini firmati

***195 Routine di interrupt in Basic** - Il Basic del C-128 non consente di mandare automaticamente in esecuzione a intervalli di tempo regolari una subroutine Basic. Esiste tuttavia la possibilità d'interrompere il programma principale e mandare in esecuzione una routine di gestione ogniquale volta si verifica una collisione tra sprite. Possiamo dunque sfruttare questo stratagemma per creare una sorgente di interrupt regolari da Basic.

```
10 FOR I=DEC("E00") TO DEC("E7E") : POKE I,0 : NEXT
40 POKE DEC("E00"),255 : POKE DEC("E03"),255
50 POKE DEC("E40"),255 : POKE DEC("E43"),255
60 SPRITE 1,1,1,1,0,0,0 : SPRITE 2,1,1,1,0,0,0
70 MOVSPR 1,0,0 : MOVSPR 2,0,100 : MOVSPR 1,0#6
90 COLLISION 1,1000
100 FOR I=1 TO 10000 : PRINT I : NEXT
...
1000 CHAR 1,20,0,"ORE "+TIME$+CHR$(7)+CHR$(13)
1010 RETURN
```

Il programma definisce due sprite a forma di barretta orizzontale, poi li colloca sullo schermo allineati verticalmente. A questo punto ne mette in movimento uno e inizia a rilevare le collisioni tra i due sprite. Il loop principale continua a visualizzare numeri crescenti mentre, attraverso l'interrupt, ogni due secondi viene visualizzata l'ora e viene emesso un suono.

Naturalmente gli sprite vengono collocati nell'area non visibile dello schermo per mascherare l'artificio, ma è possibile portarli dentro per esaminare il funzionamento della routine di rivelazione delle collisioni. Questo esame è necessario quando si desidera variare la cadenza degli interrupt: infatti alterando la velocità di scorrimento del primo sprite (riga 70), si modifica il tempo che intercorre tra un interrupt e il successivo. Se però questo tempo è troppo alto o troppo basso, può succedere che certi passaggi non vengano rilevati, o che si generino due interrupt per volta. Per risolvere tale inconveniente bisogna modificare lo spessore dei due sprite (righe 40-50), oppure calcolare accuratamente il tempo impiegato per l'esecuzione della routine di gestione.

Commodore Amiga

***196 Estensioni al C standard** - Com'è noto, il linguaggio C permette di creare strutture di dati e di nidificarle fra loro in qualsiasi numero di livelli. Tuttavia, non prevede specifiche istruzioni per gestire più facilmente questi livelli di profondità. Accade infatti che quando una struttura ne contiene al suo interno un'altra, e questa ne contiene altre ancora, fare riferimento a un

particolare elemento della struttura di livello più profondo costringe a scrivere istruzioni piuttosto prolisse, e questo alla lunga diventa seccante. Per semplificarne la gestione abbiamo escogitato un trucchetto che prende spunto dall'istruzione WITH del Pascal e ci evita di ripetere molte volte i lunghi percorsi di nidificazione delle strutture.

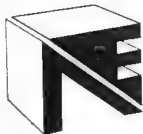
```
struct MsgPort {
    struct Node mp_Node;
    UBYTE mp_Flags;
    UBYTE mp_SigBit;
    struct Task *mp_SigTask;
    struct List mp_MsgList;
} Porta;
...
Porta.mp_Node.ln_Pri=-10;
Porta.mp_Node.ln_Name="Nome";
...
```

In questo esempio, che potrebbe far parte della routine usata da un task per creare una message port, dobbiamo accedere a due membri della struttura Node contenuta nella struttura MsgPort. Per evitare di ripetere più volte la parte invariante del percorso, si può usare la seguente coppia di metaistruzioni #define:

```
#define Do(x) With,x
struct MsgPort {
    struct Node mp_Node;
    UBYTE mp_Flags;
    UBYTE mp_SigBit;
    struct Task *mp_SigTask;
    struct List mp_MsgList;
} Porta;
...
#define With Porta.mp_Node
Do(ln_Pri)=-10;
Do(ln_Name)="Nome";
...
```

La prima #define andrà sempre posta all'inizio del programma; la seconda deve invece essere inserita subito prima delle istruzioni che ne faranno uso. Il costrutto Do(elemento) verrà espanso dal preprocessore in With.elemento, con With uguale al percorso che ci interessa. Si noti che nell'Amiga più che in altri computer ci si ritrova a utilizzare spesso gli stessi percorsi.

Un paio di particolari tecnici: la ridefinizione continua della parola With darà luogo a messaggi di warning da parte del compilatore, ma questo non porta nessun inconveniente; al limite, è possibile precedere ogni direttiva #define successiva alla prima con la direttiva #undef. Inoltre, se avete bisogno d'individuare un parametro di una struttura non contenuta ma puntata da un elemento di un'altra, basta sostituire l'operatore "." con la freccetta ("->").



NEWEL srl
computers ed accessori

A CASA TUA DIRETTAMENTE
02/33000036

Aperto nei giorni feriali dalle 9.00 alle 12.30 e dalle 15.00 alle 19.00
e il sabato dalle 9.30 alle 13.00 e dalle 14.30 alle 18.30 - chiuso il lunedì

AMIGASHOP

UNICA SEDE: VIA MAC MAHON, 75 - 20155 MILANO

Tel.: 02/323492 solo per negozio e informazioni relative acquisti in Milano - direttamente in sede Tel. 02/33000036 per ordinazioni da tutta Italia;
Fax 02/33000035 in funzione 24 ore su 24 - BBS MODEM 02/3270226 (banca dati) al pomeriggio dopo le 13.00 fino al mattino successivo

VIDEON

Basta con i noiosi filtri per i vari passaggi... Ora c'è VIDEON il VIDEON è un digitalizzatore video a colori dotato di un convertitore PAL-RGB con una banda passante di 15 KHz per ottenere immagini a colori dalle stupefacenti qualità... Funziona in risoluzioni di: 320x256 - 320x512 - 640x256 - 640x512. Può essere collegato a una qualsiasi fonte video PAL, ad esempio videoregistratori, computer, telecamere, televisori, ecc. Il prodotto permette di visualizzare il segnale video collegato all'apparecchio e in più permette la regolazione di luminosità, colore, saturazione, contrasto. È corredato di software che permette la manipolazione di immagini FF HOLD MODIFY da 32 a 4096 colori con tecniche di SURFACE-MAPPING su solidi geometrici.

L. 420.000

MINI GEN

MINI-GEN una grande novità per professionisti ed entusiasti, per ottenere sovrapposizioni di animazioni, titoli, messaggi ecc. Funziona con tutti gli Amiga ed è compatibile con programmi come TV-Text, Pro video e molti altri. Ora la videotitolazione è alla portata di tutti, semplicissimo da usare.

L. 339.000

AMIGA MODEM 2400 PAK

Modem dedicato per A500 - A1000 - A2000, esterno 300, 1200, 2400 baud (V21-22-22BIS). Autodial, autoanswer, Hayes compatibile, completo di software e cavo di connessione al computer (disponibili altre versioni, 300/1200 e 300/1200-1200/75 Videotel).

L. 339.000

ESPANSIONI DI MEMORIA AMIGA

A501 Espansione originale Commodore che porta a 1 Mega il tuo A500

L. 319.000

Amiga PROFEX Esp. esterna da 2MB, autoconfigurante, con switch on/off per A500

L. 1.290.000

Amiga 1000 RAM Esp. da 2MB per A1000 esterna autoconfigurante

L. 1.290.000

Amiga 2000 RAM Esp. interna da 2 MB, originale Commodore

L. telefonare

Tutte le espansioni sono fornite complete di chip ram e garanzia di 12 mesi!

TASTIERA MIDI per Amiga YAMAHA + int. MIDI professionale

L. 299.000

SPEDIZIONI A MEZZO CORRIERE ESPRESSO CONTRASSEGNO © PER POSTA IN TUTTA ITALIA. PAGAMENTI RATEALI BANCOPAT IN SEDE

AMIGA ACCESSORI IN OFFERTA

Drive 3" 1/2 esterno per Amiga slimline passante L. 212.000

Drive 3" 1/2 come il precedente più disconnect incorporato L. 239.000

Drive 3" 1/2 interno per A2000 NEC (con viti ecc.) L. 179.000

Drive 5" 1/4 esterno novità (AmigaDOS + MS-DOS) L. 329.000

Drive 5" 1/4 OC-118 drive per C-64 o Amiga più emulatore L. 249.000

Disponibili anche i nuovi AMIGA DRIVE NEWEL con Display Track

Interfaccia midi professionale per Amiga L. 79.000

Pro Sound Designer Gold vers. dig. audio stereo!!! L. 169.000

Scheda Janus XT per Amiga 2000 per la comp. MS-DOS L. 980.000

Scheda Janus AT per Amiga 2000 per la comp. MS-DOS L. 1.750.000

Le schede sono complete di disk drive 5" 1/4 e manuali + software

KICKSTART 1.3 ROM

Il nuovo sistema operativo dell'Amiga ora in ROM applicabile facilmente su A500 e A2000 senza saldature e senza perdere il vecchio 1.2. Disponibile anche l'inverso per chi possiede 1.3 e vuole 1.2, con interruttore per selezionarlo.

NOVITÀ:
KICKSTART in ROM + orologio per A1000 esterno L. 119.000

BOOTSELECTOR

Trasforma il secondo drive (df1:) in (df0:), evitando così l'eccessiva usura del medesimo. Risolve spesso molti problemi di caricamento dovuti alle precarie condizioni del drive interno dopo un uso frequente, semplice da installare (non necessita di saldature), ist. in italiano.

L. 23.000

DISCONNECT

Per sconnettere il secondo drive senza dover spegnere il computer, basta agire su un apposito interruttore, recuperando così memoria, di cui molti programmi spesso necessitano per un corretto funzionamento.

L. 23.000

ANTIRAM

Questo kit sconnette tutte le espansioni di memoria su Amiga, sia interne sia esterne, risolvendo anche quei problemi di incompatibilità con il software. Semplice installazione, ist. in italiano.

L. 23.000

OFFERTA!

BOOTSELECTOR + DISCONNECT + ANTIRAM L. 60.000

AMIGA MOUSE

Finalmente disponibile il mouse di ricambio originale Commodore, dedicato per Amiga 500/1000/2000.

L. 89.000

VIRUS DETECTOR PLUS L. 35.000

VIRUS DETECTOR PLUS + VIRUSKILLER software L. 45.000

Utilissimo per i noiosi e nocivi virus che si diffondono facilmente, questo dispositivo hardware è in grado di segnalare accuratamente quando un virus va a scrivere su un disco, evitando così che il virus vi rechi dei danni. Molto utile per tutti gli Amiga, si attacca alla porta drive o al drive.

AMIGA MOVIOIA

Eccezionale novità: permette di rallentare un gioco da 100 fino a 0, per poter superare tutti gli ostacoli e capire con calma il gioco. Molto utile anche per programmi grafici, animazioni, cad, ecc.; può variare la velocità d'esecuzione. Cartuccia completa di istr. in italiano.

L. 79.000

AMIGA EPROM PROGRAMMER

Nuovo programmatore di eprom per Amiga, si collega semplicemente alla porta parallela dell'Amiga e permette di programmare tutte le eprom della 2716 alle 27517 e 27011, il tutto completo di software di gestione con lettura/scrittura e verifica delle eprom. Molte opzioni come prog. veloce tramite algoritmi ecc. Semplice da usare, completo di istruzioni per l'uso.

L. 229.000

REALTIME GRABBER AMIGA

Digitalizzatore in tempo reale, in b/n per digitalizzare immagini provenienti da una qualsiasi fonte video senza bisogno di avere un fermo immagine. Risultati eccezionali a livello fotografico, predisposto per lo splitter.

L. 599.000

AMIGA SPLITTER NEWEL (RGB/PAL converter)

Per chi possiede già un digitalizzatore video del tipo Amiga Eye, Amiga Vid, Easy View, Digi View, ecc. Evita il passaggio dei tre noiosi filtri. Lo splitter Newel converte direttamente l'immagine a colori. Indispensabile per chi possiede un digitalizzatore normale: chi possiede un VD Amiga, Grabber, Realltime VD o un qualsiasi digitalizzatore in tempo reale in b/n, con Newel splitter potrà ottenere risultati straordinari.

L. 285.000

AMIGA BOX TRANSFORMER

Il famoso box d'espansione "BIG BLUE" ora è disponibile nella nuova versione per trasformare un Amiga 500/1000 in Amiga 2000. Si può così risolvere il problema delle espansioni di memoria e delle schede XT e AT per l'emulazione MS-DOS. Questo cabinet è predisposto già per 2 floppy da 3" 1/2, 1 floppy 5" 1/4, 1 hard disk + 3 slot in AmigaDOS, 3 slot IBM XT compatibili, 3 slot IBM AT compatibili, 1 slot per scheda velocizzatrice 68020/68881. Potrete quindi utilizzare tutte le periferiche dell'Amiga 2000.

L. 399.000

AMIGA FAX

Straordinario fax per Amiga, permette di inviare e ricevere segnali fax, cartine ecc. Completo di hardware di gestione, disco e manuale in italiano. L'installazione e l'uso sono di una semplicità estrema. L. 199.000

CMi ACCELERATION BOARD

L. 499.000

CMi COPROCESSOR - 68881

L. 299.000

Schede acceleratrici per Amiga 500/1000/2000. Raddoppiano la velocità del tuo Amiga portandolo a 16 MHz; molto utile per chi usa programmi grafici con Videospace, Script, Video Effect, Pro Video e molti altri. Predisposto per coprocessore matematico 68881. Metti il turbo al tuo Amiga!

HARD DISK esterno per Amiga 500 20 MB

in offerta L. 990.000

HARD DISK Amiga CARD 20 MB per Amiga 2000

L. 990.000

GVP HARD DISK con AUTOBOOT per Amiga 500 (Fast File System DNA)

L. telefonare

GVP HARD CARD con AUTOBOOT 20 MB con controller (Fast File System DNA)

L. 1.390.000

GVP HARD CARD con AUTOBOOT 40 MB con controller (Fast File System DNA)

L. 1.690.000

GVP HARD CARD con AUTOBOOT + espansione 2MB (Prodrive) + hard quantum 45MB 11ms.

L. 2.890.000

HARD DISK CARD per Amiga 2000 (con schede XT-AT partizionabili)

20MB.....L. 619.000

32MB.....L. 759.000

40MB.....L. 939.000

★
OFFERTA SPECIALE
★

I NOSTRI DISK DRIVE

DISK DRIVE SLIM MECCANICA NEC BEIGE

sono disponibili:

per Amiga 500 3,5 pollici passante compreso disconnect L. 239.000

per Amiga 500 5,25 pollici 40/80 tracce passante L. 350.000

per Amiga 2000 interno L. 179.000

per C-64 OCC118 L. 239.000

L'HARD DISK A590: 20 MB PIÙ ESPANSIONE RAM PER L'AMIGA 500

Un hard disk da 20 MB creato su misura per il Commodore Amiga: prestazioni decisamente superiori rispetto a un disk drive e un'espansione RAM on-board che può arrivare fino a 2 MB

di Dario Greggio

Al momento dell'uscita sul mercato della macchina, il disk drive da 880K compreso nel bassissimo prezzo dell'Amiga 500 sembrava in grado di soddisfare qualsiasi esigenza. Invece, la realizzazione di programmi sempre più complessi che richiedono enormi quantità di dati, ha dato origine a poco a poco a una schiera di utenti in attesa... attesa che il sistema completi il boot, attesa per esaminare una semplice directory, attesa che un programma venga caricato in memoria o che le librerie linked vengano completamente digerite dal compilatore. Per non parlare poi dei casi in cui per il caricamento di un programma bisogna usare più di un disco.

Forse non ci si accontenta mai, ma è innegabile che lo stupendo disco da tre pollici e mezzo si è rivelato a lungo andare un collo di bottiglia per il computer, un dispositivo dalle prestazioni enormemente inferiori alla strabiliante velocità di altri elementi dell'Amiga, come i chip custom.

Con i suoi 20 MB di memoria e

un tempo di accesso medio di circa 80 millisecondi, la periferica A590 risolve molti di questi problemi con una spesa contenuta, e dà anche l'occasione di aggiungere da uno a due megabyte di espansione RAM alla memoria di sistema.



Le caratteristiche

Esteticamente questo hard disk si presenta con una linea analoga a quella del computer, anche se è molto più stretto e meno slanciato. Le sue dimensioni sono circa 12 x 30 x 5 centimetri. Va collocato sulla sinistra dell'Amiga, per

poterlo connettere al bus di espansione. A questo proposito abbiamo subito constatato un particolare positivo: la periferica dispone di un proprio alimentatore esterno la cui accensione è subordinata elettronicamente all'accensione del 500, così da evita-

re il rischio di sovraccaricare l'alimentatore del computer. In compenso, però, l'unità A590 non ripete il bus alla sua sinistra e questo impedisce l'inserimento in cascata di ulteriori periferiche. Quest'ultimo particolare ci sembra poco felice, in quanto costringe chi desidera ulteriori espansioni a utilizzare soltanto schede dotate di un bus passante sul quale collegare l'hard disk come ultimo elemento della catena. Oppure a rinunciare all'hard disk.

Rompiamo ora l'idillio tra l'Amiga 500 e l'A590 (da un punto di vista estetico il complesso dei due strumenti fa un ottimo effetto) per passare all'analisi interna. Quattro viti bloccano il coperchio superiore mentre altre quattro tengono fermo il circuito stampato e contribuiscono a fissare l'hard disk vero e proprio, di

marca Epson. Gli unici connettori elettrici da scollegare per isolare il coperchio dalla base sono quelli relativi ai led e alla ventola di raffreddamento. Su un circuito stampato grande quanto l'involucro esterno trovano posto la logica di controllo della memoria di massa, gli zoccoli che accolgono i chip di espansione della memoria RAM, e il connettore che s'innesta nella porta d'espansione del 500. Il cablaggio è abbastanza ordinato, per quanto siano da criticare l'assenza di componenti in tecnologia SMD (il che denota una progettazione non particolarmente moderna, ma evidentemente più economica) e un collegamento volante su un piedino di un integrato TTL 7474. I chip impiegati per l'espansione RAM sono i recenti M514256-12. L'hard disk vero e proprio, che occupa lo spazio fisico di un normale disk drive da 3,5", è saldamente fissato al circuito stampato tramite alcune viti, e la connessione elettrica con la logica di controllo avviene tramite un flat cable da 40 fili.

Sul retro troviamo poi il connettore per l'alimentazione esterna, una serie di quattro DIP switch che permettono di abilitare/disabilitare l'autoboot da hard disk e d'indicare il numero di periferiche collegate in cascata, e infine un connettore DB-25 femmina che fa capo a una porta SCSI (Small Computer System Interface), una parziale soluzione alla mancanza di duplicazione del bus. Dopo il Macintosh, questa interfaccia che consente la creazione di semplici e funzionali catene di dispositivi è diventata praticamente uno standard nel campo delle memorie di massa.

Per esempio, una porta SCSI può essere collegata a un disk drive, di qui a un secondo disk drive e infine a un hard disk; trattandosi di un'interfaccia parallela, i dati possono viaggiare sui suoi collegamenti fino alla velocità di quattro milioni di byte al secondo.

Di fronte a unità a disco fisso "portatili" come questa, viene da chiedersi se le testine possano incontrare problemi durante il trasporto. A questo scopo, il software della release 1.3 installato sull'hard disk contiene un comando, SHIPDISK, che viene utilizzato per "parcheggiare" le testine in una zona sicura, operazione indispensabile per qualunque trasporto dell'unità.



Per dovere di cronaca, citiamo infine un problemuccio che abbiamo riscontrato appena acceso l'Amiga 500: la ventola di raffreddamento montata all'interno dell'A590, prima ventola a entrare ufficialmente nella vita di un utente dell'Amiga 500, è risultata molto rumorosa e oltretutto soffiava al contrario, dall'esterno verso l'interno dell'unità!

Ovviamente, i tecnici della Commodore ci hanno subito garantito che il problema riguardava solo quel particolare esemplare, e che piccoli difetti d'assemblaggio come questo, sempre possibili in un prototipo, sarebbero stati completamente risolti prima d'immettere sul mercato la versione finale della periferica.

Il funzionamento

Prima di parlare delle nostre impressioni sull'unità A590, precisiamo che la prova è stata condotta su un esemplare per ora unico di Amiga 500, che offriva on-board l'espansione di memoria A501. Questo esemplare sarà probabilmente la base per un nuovo modello di Amiga 500 che dovrebbe contenere anche il nuovo ECS (Enhanced Chip Set). Ricordiamo che l'ECS è una nuova serie di chip custom destinata a sostituire gli attuali Agnus, Paula, e Denise. Le prime notizie sui nuovi chip, talvolta incerte e contraddittorie, hanno creato una certa confusione negli utenti.

Apriamo quindi una breve parentesi che risolve almeno i dubbi maggiori.

La prima caratteristica è che la nuova serie sarà in grado di accedere a 1 MB di RAM, contro i 512K attuali; questo consentirà di gestire un maggior numero di schermi e finestre per quanto

riguarda le applicazioni grafiche, e di suoni campionati per quanto riguarda le applicazioni che impiegano i canali audio. Ovviamente, la stessa mappa di memoria dell'Amiga dovrà essere modificata per ottenere un megabyte di chip RAM continuo; attualmente infatti i 512K di chip RAM terminano all'indirizzo 0x080000, mentre i 512K di fast RAM in dotazione all'Amiga 2000 e opzionali sul 500 iniziano all'indirizzo 0xC00000, e questa configurazione discontinua non permetterebbe all'ECS di accedere a un unico blocco di RAM da 1MB.

La seconda caratteristica di rilievo è che l'ECS permetterà risoluzioni di schermo superiori all'attuale limite di 640 x 512 pixel.

La risoluzione orizzontale potrà arrivare fino a 1280 pixel, che ovviamente risulteranno nella giusta nitidezza se il monitor di cui si dispone ha una risoluzione orizzontale almeno dello stesso livello (meglio se superiore). A questo proposito occorre precisare che i monitor 1084 e 2080, avendo una risoluzione orizzontale di 640 punti, non riusciranno a visualizzare con la giusta nitidezza le immagini da 1280 pixel orizzontali che l'ECS consentirà di creare. Per quanto riguarda la risoluzione verticale, l'ECS consentirà di visualizzare 512 linee per schermo senza bisogno dell'interlace. Ricordiamo che l'interlace è un sistema per creare schermi da 512 linee (anziché da 256) completando ogni schermo con due passaggi del cannone elettronico spostati di mezza linea l'uno rispetto all'altro. Il problema è che in questo modo i 50 quadri al secondo diventano in pratica 25, troppo pochi per sfruttare al meglio le proprietà di persistenza della retina umana. Il risultato è il cosiddetto flickering (sfarfallio).

Per ottenere un buon risultato senza l'interlace occorre disporre di un monitor multisync, ovvero un monitor in grado di visualizzare un maggior numero di linee per quadro nello stesso periodo di tempo (1/50 di secondo). La risoluzione verticale di questi monitor non supererà comunque le 512 linee.

Sull'ECS si può aggiungere ancora che i nuovi chip custom non saranno installabili nell'Amiga 1000, ma conviene rimandare ogni ulteriore approfondimento a quando la Commodore li immetterà sul mercato.

Passiamo finalmente ad analizzare le caratteristiche della nuova periferica: per quel che riguarda l'espansione RAM, occorre anzitutto precisare che sul circuito

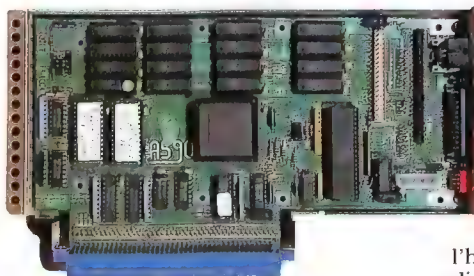
stampato dell'A590 sono presenti gli zoccoli per installare on-board 2 MB di RAM, ma che tale memoria non verrà distribuita dalla Commodore: saranno i singoli utenti a decidere quanta memoria aggiungere, procurandosi sul mercato i necessari integrati RAM



L'hard disk da 20 megabyte nell'A590

da 4 x 256 kilobit. Considerando gli elevati costi delle memorie RAM, ci sembra una buona scelta lasciare all'acquirente la possibilità di valutare l'investimento.

Dal punto di vista del funzionamento, non c'è molto da dire, in quanto il sistema utilizza la memoria RAM aggiunta nell'A590 senza fare alcuna distinzione con la memoria già presente. Nel



La scheda di controllo con 2 MB installati

modello da noi provato, la RAM era stata estesa al limite massimo di 2 MB, facendo comparire l'eloquente dato "Fast: 2560" nella finestra relativa all'utility *mcl*k che abbiamo mandato in esecuzione. Naturalmente questa memoria aggiuntiva lavora alla frequenza di 7.09 MHz come il resto del computer e non si tratta di

memoria gestibile con un bus dati da 32 bit. Del resto la frequenza di 14 MHz non sarebbe stata comunque raggiungibile... avevate fatto caso alla presenza dei chip M514256-12 da 120 nanosecondi?

Da parte nostra abbiamo comunque svolto qualche analisi nei meandri del sistema, facendo uso dell'"esaminatore di liste" che accompagnava nel numero scorso l'articolo "ExecBase: i parametri dinamici e le liste di sistema"; abbiamo così scoperto come vengono viste dal sistema operativo le periferiche appena installate. La RAM eventualmente aggiunta viene

inserita nella lista dei chunk di memoria disponibili con il nome Expansion. In questo modo la memoria dell'espansione si distingue sia dalla memoria chip sia da quella fast, gli unici tipi di memoria presenti in un Amiga 2000 standard. La distinzione non crea comunque problemi per i programmatori, dal momento che i 2 MB di espansione ai fini della gestione della memoria vengono considerati dal sistema come RAM di tipo fast.

Per quanto riguarda l'hard disk, invece, abbiamo constatato la presenza nella lista di sistema DeviceList di due nuovi dispositivi di I/O, *scsi.device* e *xt.device*, che dovrebbero essere delegati alla gestione della porta parallela ad alta velocità e dell'hard disk fisico. Sia chiaro che l'hard disk non è gestito dal dispositivo TrackDisk, che si occupa soltanto delle unità a floppy disk. Di conseguenza, pur installando l'A590, è sempre possibile collegare in cascata tre disk drive esterni.

È interessante a questo punto analizzare qualche caratteristica dell'hard disk. Ci ha colpito molto favorevolmente il boot di sistema seguito da disco fisso, reso possibile dalla recente versione 1.3 del *Kickstart*, peraltro indispensabile perché l'A590 possa funzionare:

bastano pochi secondi e le icone prendono forma sullo schermo del *Workbench*. Va notato però che se è presente un disco nel disk drive DF0:, il boot viene eseguito da quel disco e non dall'hard disk. Lo stesso accade se l'autoboot da hard disk è stato disabilitato agendo sui DIP switch.

L'esplorazione dei vari livelli di subdirectory avviene molto rapidamente e non si rimpiangono certo i vecchi dischi da 3,5". Tutto questo senza aver provato ad affrontare una situazione di lavoro vera e propria, nella quale fosse determinante anche la quantità di memoria a disposizione. Siamo comunque convinti che i vantaggi in questo caso sarebbero stati (e saranno) ancora maggiori.

Abbiamo eseguito alcuni test per avere un'idea più precisa della velocità di questo hard disk, e i risultati si possono vedere nella Tavola 1. La prova dell'A590, trattandosi di un prodotto non ancora disponibile per il pubblico, è avvenuta presso la Commodore, e quindi non è stato possibile creare opportuni programmi di benchmark in C, cercando magari di accedere direttamente al dispositivo di I/O xt.device. Abbiamo quindi ripiegato su un test scritto in Amiga BASIC, riservandoci future verifiche quando riceveremo il prodotto in redazione. Anche se il controllo da Basic della scrittura/lettura dei caratteri in un file è meno diretto che con il C, va comunque detto che le indicazioni di massima di cui avevamo bisogno le abbiamo ottenute con molta facilità (si noti che per il sistema l'hard disk porta il nome "EHD0:").

Dai dati ottenuti si nota che i miglioramenti nella velocità d'accesso in scrittura rispetto alla stessa prova condotta su floppy disk sono dell'ordine del 30-35 per cento, mentre in lettura non si supera il 10 per cento. Questa differenza si spiega andando ad analizzare più da vicino il lavoro svolto a livello di sistema. Durante l'accesso in scrittura a un disco (floppy disk o hard disk) il DOS

deve prima di tutto riservare lo spazio per l'header di un nuovo file (cancellando un eventuale file che abbia lo stesso nome); in seguito deve verificare per ogni blocco di dati se è presente sul disco un settore libero. Se l'esito è

adiacenti (a meno che non si abbia a che fare con file piuttosto frammentati, il che può accadere se sono stati sovrascritti diverse volte). Si comprende, dunque, che la velocità di accesso alla singola posizione sul supporto

Tavola 1

Blocco da 60K	A590	disk drive	RAM disk
Scrittura	25,5 s	39 s	22 s
Lettura	32 s	34 s	28 s
Blocco da 120K	A590	disk drive	RAM disk
Scrittura	49 s	75 s	44 s
Lettura	63 s	68 s	58 s

positivo lo alloca, e vi memorizza il blocco di dati. A parte l'ultima azione, il precedente lavoro di scansione del supporto magnetico (floppy o hard che sia) è paragonabile come tempi a quello svolto

magnetico, che è notevolmente superiore nel caso di un disco fisso, influisce molto di più nella scrittura che nella lettura.

L'attendibilità dei test effettuati, per quanto semplici, è stata

Listato 1

```
print time$
open "EHD0:pippo" for output as 1
for i=1 to 10000
print #1,"pippo"
next i
close 1
print time$
```

Listato 2

```
print time$
open "EHD0:pippo" for input as 1
for i=1 to 10000
input #1,a$
next i
close 1
print time$
```

Nota: questi sono i programmi base utilizzati nella prova. Di volta in volta abbiamo alterato l'identificatore della periferica e il numero d'iterazioni del loop.

per l'effettiva scrittura dei byte nel settore. Ovviamente durante la rilettura non vi sono invece altri lavori da svolgere se non quello di leggere i byte dai diversi settori del disco, i quali di solito si trovano in posizioni fisicamente

confermata quando li abbiamo applicati al magnifico hard disk offerto (speriamo presto) insieme al sistema operativo Unix System V, presentato su questo stesso numero di *Commodore Gazette*: la strabiliante velocità di quella peri-

ferica (18 ms di tempo d'accesso medio) è risultata perfettamente coerente con i tempi che abbiamo rilevato.

I tecnici della Commodore non hanno fornito dati certi sul fattore di interleave dell'hard disk A590, ma in base al comportamento generale della macchina si può supporre che non sia particolarmente buono. Questa impressione, unita al fatto che il tempo medio di accesso dichiarato dal costruttore è molto elevato, 80 ms, ci porta a concludere che le prestazioni di questo hard disk non sono paragonabili a quelle delle unità con tempi d'accesso variabili tra i 28 e i 40 ms che animano il mondo MS-DOS. La capacità di memoria ci sembra invece adeguata, tenendo presente che un utente medio dell'Amiga 500 dovrebbe aver bisogno di circa 4-5 MB per installarvi i suoi programmi preferiti e di un altro paio di megabyte per i dati, sempre che non si usino tantissime schermate grafiche o brani musicali campionati (in questi casi, tuttavia, non ci sembra utile riempire un hard disk soltanto per vedere le immagini formarsi sul video un po' più in fretta).

Desideriamo infine sottolineare che durante le nostre prove non era attivo il Fast File System (una tecnica software per aumentare la velocità, offerta dalla versione 1.3 del sistema operativo) che avrebbe potuto rendere difficile attribuire i meriti all'hardware o al software. Inoltre la prova ha messo in luce che la velocità di accesso del disco virtuale in RAM è superiore a quella dell'hard disk A590 e confrontabile con quella dell'hard disk che verrà distribuito nell'Amiga 2500 UX. Questo dato è incompleto, in quanto la nostra prova non mirava a sondare il funzionamento dei tre tipi di memorie di massa in ogni aspetto, ma è comunque importante per comprendere il funzionamento dei dispositivi dell'Amiga. Naturalmente, quando si accede a un disco in maniera meno sequenziale di come abbiamo fatto nei nostri test, le prestazioni del RAM disk sono come sempre molto superiori.

Conclusioni

In rapporto agli hard disk di cui si è parlato nel numero 2/88 di

Commodore Gazette, possiamo dire che le caratteristiche dell'A590 sono inferiori a quelle delle periferiche Xebec e Commodore viste in quell'occasione, ma che esiste anche una marcata differenza di prezzo in favore dell'A590. Il prezzo di listino è stato infatti fissato in lire 995.000 (Iva esclusa) per la versione che la Commodore sembra avere intenzione di distribuire, ovvero quella senza la RAM aggiuntiva. Si tratta di una cifra che può comunque sembrare alta in assoluto, ma dopotutto il rapporto di prezzi tra l'A500 e l'A590 è lo stesso che c'era tra il C-64 e il disk drive 1541.

Il giudizio su questa periferica è complessivamente positivo: la velocità non altissima, infatti, è controbilanciata dal prezzo abbordabile e dall'importante possibilità di aggiungere fino a 2 MB di RAM all'Amiga 500. ■

**Per ulteriori informazioni
contattare direttamente:**

Commodore Italiana spa
(A590, L. 995.000 Iva esclusa)
Via F.lli Gracchi, 48
20092 Cinisello Balsamo (MI)
(Tel. 02/618321)

R I C E R C A D I P E R S O N A L E

**LA IHT GRUPPO EDITORIALE CERCA UN REDATTORE TECNICO
PER LA DIVISIONE LIBRI E LA RIVISTA COMMODORE GAZETTE**



SI RICHIEDE:
Buona conoscenza dei principali linguaggi di programmazione
Buona conoscenza dell'inglese tecnico
Interesse per i computer Commodore
Buona capacità di esposizione scritta

SONO RITENUTI TITOLI PREFERENZIALI:
Laurea in scienze dell'informazione
Conoscenza approfondita dei computer
Precedenti esperienze nel mondo editoriale

L'assunzione è a tempo pieno. La località di lavoro è Milano.
Gli interessati possono inviare un curriculum dettagliato a:
IHT Gruppo Editoriale - Via Monte Napoleone, 9 - 20121 Milano

UN COMPUTER, UN LIBRO...

...LE ALI

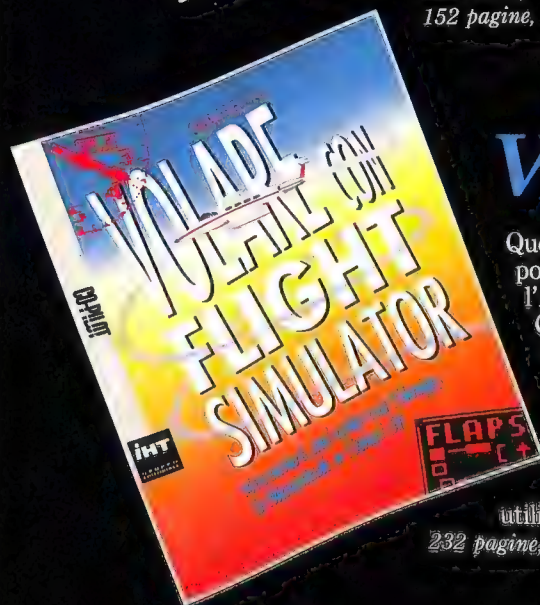


F LIGHT SIMULATOR CO-PILOT

Questo volume è un vero "istruttore di volo" per coloro che possiedono il programma Flight Simulator della Microsoft per i PC IBM e compatibili, oppure Flight Simulator II della SubLogic per Commodore 64, Apple II, Atari 800 XL e XE.

Non è un semplice compendio di comandi da ricordare a memoria, ma un brillante compagno d'avventura, scritto con estro e ironia. Un libro da sfogliare con il computer acceso, per imparare passo dopo passo i principi del volo "rettilineo e livellato", le procedure di decollo e di atterraggio, del volo strumentale e notturno, e del volo acrobatico.

152 pagine, L. 30.000



V OLARE CON FLIGHT SIMULATOR

Questo volume si rivolge a tutti gli utenti che possiedono la versione di Flight Simulator per l'Amiga, l'Atari ST o l'Apple Macintosh.

Con uno stile spigliato, ma con grande attenzione a ogni dettaglio, l'autore conduce l'allievo pilota attraverso le varie fasi di un corso di volo che si concluderà con un primo approccio al volo acrobatico. Il testo è completato da dettagliate cartine di rotta, schemi che illustrano l'uso dei comandi, parametri di volo ideali e procedure

utili nelle più svariate situazioni di volo.

232 pagine, L. 45.000

68020 E SISTEMA OPERATIVO UNIX PER L'AMIGA 2000

Con il 68020 e il coprocessore matematico della scheda A2620, un hard disk da 80 MB, e uno streamer da 150 MB nasce l'Amiga 2500 UX, una workstation Unix davvero competitiva

di Dario Greggio

Il grande balzo in avanti è finalmente possibile. La Commodore ha deciso che per rendere l'Amiga 2000 più interessante agli occhi dell'utenza professionale è necessario spingersi oltre la compatibilità MS-DOS offerta dalle schede A2088 e A2286. E per raggiungere questo scopo è stato realizzato un A2000 configurato in maniera tale da mettere a disposizione dell'utenza Commodore una workstation Unix davvero competitiva.

La novità ci è stata presentata in anteprima assoluta nella sede della Commodore. Gli elementi di cui è composto il sistema sono una scheda A2620 installata nel CPU slot dell'Amiga 2000, un superbo hard disk da 80 MB gestito dal controller A2090A, uno streamer da 150 MB e una versione parzialmente personalizzata del sistema operativo Unix System V.

A questo punto è utile sottolineare che tutte le considerazioni relative alla scheda acceleratrice A2620 sono di particolare interesse anche per chi desidera migliorare le prestazioni della propria macchina senza arrivare a costruirsi una workstation Unix, perché la scheda viene distribuita anche come prodotto singolo.

Abbiamo avuto a disposizione per una giornata il sistema (ribattezzato con il nome Amiga 2500 UX, ma trattandosi di un prototipo nulla è da considerarsi definitivo) e anche se la prova non ha potuto essere particolarmente approfondita siamo ugualmente riusciti a ottenere informazioni e dati interessanti. Promettendo di tornare al più presto sull'argomento, entriamo nel vivo di questa anteprima iniziando dalle ragioni che hanno spinto la Commodore a entrare nel mondo Unix, per poi scoprire nei dettagli come l'Amiga 2000 è stato trasformato in una completa workstation.

Perché lo Unix

In occasione della visita alla Commodore Italiana si è discusso delle nuove strategie pianificate dall'azienda per aumentare la sua presenza nel settore professionale. L'intenzione è di rafforzare ulteriormente la già affermata linea PC, e contemporaneamente di aprire la strada a sistemi professionali diversi dagli IBM compatibili. L'unica macchina su cui la Commodore può lavorare in questo senso è l'Amiga 2000, un

computer che però costituisce un ibrido a cavallo fra una macchina professionale e un home computer, per quanto possa essere dotato della scheda A2620 e della compatibilità MS-DOS. Cerchiamo di capire il perché.

La velocità e la potenza di calcolo che si ottengono con la scheda A2620 o con altre schede acceleratrici trasformano l'Amiga in un computer dalle prestazioni sicuramente confrontabili con quelle di macchine assai più costose, ma non bastano a renderlo veramente competitivo nel difficile mercato professionale. Il neo dell'Amiga rimane infatti il sistema operativo che, per quanto all'avanguardia (pari, se non superiore, a quello di un Macintosh e precursore di sistemi operativi come l'OS/2), non rappresenta uno standard commerciale al di fuori del mondo Commodore ed è troppo giovane per essere del tutto esente da difetti: due ragioni che non hanno favorito la sua diffusione nei settori professionali e quindi neanche un'elevata produzione di software applicativi.

Per contrapporsi con successo alle macchine dotate di processori Intel, occorre almeno un Motorola 68020 e un sistema operativo

affidabile e diffuso nel mondo professionale: in pratica, un Amiga corredato della scheda A2620 e del sistema operativo Unix.

Per quanto abbia ormai festeggiato il suo ventesimo anniversario, il sistema operativo Unix è tuttora ritenuto in tutto il mondo una delle scelte migliori per affrontare esigenze professionali. È un sistema nato e diffuso soprattutto nelle università, studiato appositamente per agevolare il lavoro dei gruppi di ricerca, e offre quindi un'ottima gestione multiutente e un'utilissima condizione delle risorse di sistema.

La prima versione del sistema operativo Unix, realizzata per conto del Computing Science Research Group, venne presentata nel 1969 da Ken Thompson e Dennis Ritchie, due ricercatori dei laboratori Bell (New Jersey, USA). Per molti anni le varie versioni del programma restarono ben lontane dai canali commerciali, e vennero utilizzate principalmente dalle università e dai grandi gruppi di ricerca. Erano versioni create per grandi sistemi come il DEC PDP-11, la serie DEC VAX e il sistema IBM 370. Soltanto nel 1975 venne studiata una versione in C installabile con opportuni ritocchi su macchine molto diverse dal punto di vista hardware. Questo passo costituì il trampolino di lancio che nel 1981 proiettò lo Unix nei canali commerciali, grazie soprattutto all'adattamento per computer dotati di microprocessori come il Motorola 68000, macchine evidentemente più piccole ma molto più diffuse.

Oggi come oggi, lo Unix è ritenuto il sistema operativo più affidabile in assoluto. È diventato praticamente uno standard, soprattutto per quanto riguarda gli ambienti di sviluppo del software, tant'è vero che molti programmi vengono creati in sistemi Unix anche se poi dovranno essere

adattati a contesti diversi (si pensi a GEOS, esempio in formato ridotto di cross-development per il C-64, oppure ai noti programmi di ray tracing realizzati originariamente in ambiente Unix e oggi disponibili per l'Amiga).

Ma per lanciare una macchina Unix sul mercato, sebbene con un prezzo molto inferiore rispetto alle sue dirette concorrenti, non basta metterla nelle vetrine dei negozi. È necessario anche garantire assistenza e competenza nei punti vendita, un provvedimento che comunque andrà a rafforzare tutta la linea professionale della Commodore, PC compresi. In

configurazione finale dovrebbe diventare il sistema Unix più economico in assoluto (escluso forse il futuribile NeXT). Coerentemente con la strategia illustrata, il debutto di questa macchina dovrebbe essere accompagnato da un ampio ventaglio di applicazioni professionali in ambiente Unix, come CAD e DTP. Se infatti la formula hardware più pacchettata applicativo è sempre auspicabile, in ambito Unix è indispensabile: proporre una macchina Unix senza sostenerla con l'adeguato software sarebbe assurdo, e questo la Commodore lo sa bene.



Sullo schermo i tre ambienti di lavoro dell'A2500 UX

questo senso ci sono diversi progetti per il futuro: alcuni Commodore Point, per esempio, verranno trasformati per far fronte alle maggiori esigenze del mercato professionale e per offrire la necessaria competenza, così da diventare efficienti centri di distribuzione e punti di riferimento per l'assistenza. Anche la formula di promozione non vedrà più il solo computer "chiavi in mano", ma il trinomio computer più pacchetto applicativo più periferiche, così da andare incontro alle esigenze del mercato con configurazioni complete e affidabili.

L'Amiga 2500 UX, nella sua

La scheda A2620

Passiamo ora all'esame dei vari elementi costitutivi dell'Amiga 2500 UX. La scheda A2620, il vero e proprio "cuore" della macchina, è la prima e unica scheda acceleratrice distribuita ufficialmente dalla Commodore. Si tratta quindi di un prodotto estremamente interessante, anche per chi desidera potenziare il suo A2000. Contiene infatti 2 MB di RAM a 32 bit espandibile fino a 4 MB, un coprocessore matematico 68881, una MMU (Memory Management Unit, chip di gestione della memoria) 68851, e una CPU 68020 che lavora a 14 MHz.

La realizzazione è di tipo tradizionale: non prevede soluzioni avveniristiche, e neanche l'impiego di componenti SMD (Surface Mounted Devices), mentre spiccano i tre zoccoli quadrati che sorreggono i chip. Regna ovunque l'ordine e la funzionalità... fuorché in un punto dove fanno capolino una resistenza e un filo volante.

La RAM è composta da chip 414256 ad alta densità e a montaggio verticale. Viene vista come memoria a 32 bit quando la macchina è controllata dal 68020, mentre non è utilizzabile quando

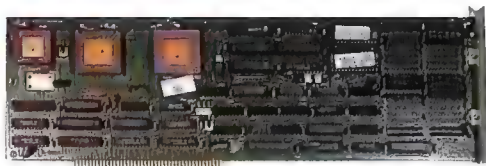
si fa funzionare l'Amiga con il suo 68000 di serie.

L'installazione della scheda, che dev'essere inserita nel CPU slot della scheda madre, non comporta interventi hardware sugli Amiga 2000B, mentre sull'Amiga 2000A sono necessarie alcune operazioni. Nella serie A, infatti, i 512K di fast RAM sono montati su una scheda di serie che occupa proprio il CPU slot; occorre quindi rimuovere questa scheda d'espansione, perdendo così 512K di memoria, e ci è stato confermato che non c'è attualmente rimedio a questo spreco hardware. Inoltre, sempre con il 2000A, sembra che saranno necessari altri piccoli interventi su ponticelli e cavallotti, mentre viene smentita dalla Commodore la notizia che sarebbe anche necessario estrarre il 68000. Abbiamo detto "sembra" perché purtroppo risposte certe su questi problemi non ne abbiamo avute.

All'accensione della macchina, o al reset, la scheda prende il sopravvento e configura il sistema in modo che venga esclusa la CPU di serie, il 68000. Agli occhi dell'utente l'Amiga si presenta quindi con una CPU 68020 funzionante a 14 MHz e 3 MB di RAM, due dei quali, quelli della scheda, accessibili tramite un bus dati da 32 bit. Questa configurazione di default non è comunque l'unica. Il firmware presente nella ROM della scheda consente infatti di sceglierne altre tenendo premuti entrambi i pulsanti del mouse durante l'accensione o il reset della macchina. In questo caso, appare una schermata costituita da tre opzioni selezionabili tramite mouse: "AmigaDOS on 68000", "AmigaDOS on 68020", "Amiga UX". La prima opzione non fa altro che rendere attivo il 68000 di serie. In questa configurazione la scheda A2620 è completamente invisibile al sistema, come se non fosse inserita; questa opzione è stata prevista per garantire la compatibilità con quei pochi pro-

grammi che non funzionerebbero con il 68020 (per esempio, con i programmi che automodificano i propri codici). La seconda opzione corrisponde alla configurazione di default che si ottiene senza premere i due pulsanti del mouse. La terza opzione, infine, consente di attivare la macchina in ambiente Unix, e funziona solo con l'Amiga UX (attivando questa opzione senza che sia presente un hard disk con una partizione Unix si provoca il blocco del sistema).

Nella configurazione di default, l'espansione RAM e il funzionamento con frequenza di clock raddoppiata offerti dalla scheda A2620 sono gli unici vantaggi ottenibili, dal momento che fino all'attuale release 1.3 il sistema operativo dell'Amiga non sa assolutamente cosa sia una MMU



La scheda acceleratrice A2620 da inserire nel CPU slot

come il 68851, né può servirsi delle nuove istruzioni della CPU 68020; il discorso è invece un po' diverso per il coprocessore matematico, perché ormai quasi tutti i pacchetti software che richiedono calcoli complessi vengono realizzati in due versioni, una che sfrutta le librerie matematiche dell'Amiga e una che sfrutta la presenza del coprocessore matematico. Si noti che l'accesso diretto al coprocessore matematico è disponibile solo con il 68020, mentre con il 68000 l'accesso può avvenire solo se il coprocessore matematico viene trattato come una periferica. Con questo genere di applicazioni, e non certo grazie al sistema operativo, la scheda A2620 può quindi essere sfruttata al massimo delle sue capacità.

Comunque, anche se il sistema

operativo dell'Amiga trae pochi vantaggi dalla presenza della scheda, le prestazioni dell'Amiga migliorano sensibilmente quando si combinano una frequenza di clock più elevata, una memoria RAM a 32 bit, e una CPU dotata di bus dati da 32 bit (come ben sa chi ha letto l'articolo sulle schede acceleratrici nel numero 6/88 di *Commodore Gazette*), tanto che si giunge in genere a raddoppiare la velocità di esecuzione dei programmi, e a quadruplicarla quando viene chiamato in causa anche il coprocessore matematico. Alcuni esempi rendono più evidente questo incremento di prestazioni. I lunghi tempi di generazione delle immagini di uno slideshow realizzato tramite il pacchetto *Sculpt 4D* (che è in grado di sfruttare il coprocessore matematico) si riducono del 60 per cento

se si installa la scheda A2620. Se s'impiega il pacchetto *Aegis Draw* nella versione realizzata per il 68020 in coppia con il 68881, si riscontrano incrementi di velocità ancora maggiori: il tempo necessario a calcolare e visualizzare un'immagine dimostrativa si rivela inferiore a un terzo del tempo

richiesto con un 68000 a 7,09 MHz (frequenza di clock dei sistemi Amiga europei).

Risultati così incoraggianti rendono la scheda A2620 quasi indispensabile per certe applicazioni, e viene da chiedersi quali traguardi si potrebbero raggiungere installando anche un quarzo da 25 MHz che fornisca di un nuovo clock il coprocessore matematico.

L'espansione RAM a 32 bit

Per quanto riguarda l'espansione RAM da 2 MB a 32 bit presente sulla scheda A2620, è interessante analizzare come viene vista dal sistema operativo dell'Amiga. Per affrontare il discorso occorre una breve premessa. Come è stato spiegato nell'articolo del numero

scorso, "ExecBase: i parametri dinamici e le liste di sistema", nell'Amiga esiste una lista di sistema, MemList, a cui si deve ricorrere per sapere quanta memoria libera c'è nel sistema e dove si trova. I suoi nodi costituiscono una serie di strutture MemHeader, ognuna delle quali si riferisce a un particolare tipo di memoria RAM e ne elenca i chunk liberi (per chunk s'intende un'area di memoria). Queste strutture MemHeader sono disposte secondo un ordine di priorità; per esempio, nell'Amiga 2000 la prima struttura MemHeader della lista elenca i chunk liberi di memoria fast e ha priorità 0, mentre la seconda elenca i chunk liberi di memoria chip e ha priorità -10. La priorità dei diversi tipi di RAM determina l'ordinamento delle corrispondenti strutture MemHeader nella lista MemList, il quale a sua volta influisce sul comportamento della funzione AllocMem, che il sistema e i task dei programmatori devono impiegare per allocare memoria. Chiamando questa funzione, oltre alla quantità di byte che si desidera allocare occorre indicare come argomento un insieme di flag che ne modificano il comportamento. Se viene indicato il flag MEMF_CHIP viene allocata memoria nella chip RAM, e quindi la citata priorità è ininfluente. Se viene indicato il flag MEMF_FAST viene allocata memoria nella fast RAM. Se viene infine indicato il flag MEMF_PUBLIC interviene la priorità: AllocMem tenta di soddisfare la richiesta iniziando dalla memoria RAM che nella lista MemList possiede la più alta priorità, e prende in considerazione i livelli a priorità inferiore solo se i precedenti non presentano chunk di memoria libera sufficientemente ampi.

Fatte queste premesse sulla gestione della memoria da parte dell'Amiga, torniamo alla nostra prova hardware. Tramite l'utility per esaminare le liste di sistema (pubblicata anch'essa nel numero scorso di *Commodore Gazette*), abbiamo osservato che i 2 MB di

RAM della scheda A2620 vengono automaticamente indicati nella lista di sistema MemList con la più alta priorità (il nodo si chiama Expansion). Con questa configurazione di default, ogni volta che viene chiamata la funzione AllocMem indicando il flag MEMF_PUBLIC, il sistema cerca sempre di allocare la memoria richiesta nella RAM a 32 bit, aumentando considerevolmente la velocità d'esecuzione dei task. Se invece per ragioni non dipendenti dal sistema la struttura MemHeader relativa alla RAM da 32 bit non dovesse trovarsi in testa alla lista, occorre estrarla, aumentarne opportunamente la priorità e reinserirla nella lista tramite la funzione Enqueue (che inserisce un nodo in una lista tenendo conto della sua priorità).

Della MMU presente sulla scheda non abbiamo parlato, perché in ambiente AmigaDOS non si prevedono impieghi per questo coprocessore: il sistema operativo non lo riconosce e il software difficilmente potrebbe avvantaggiarsene. Si tratta comunque di un dispositivo della massima importanza in sistemi più evoluti come lo Unix, dove è fondamentale salvaguardare l'integrità del sistema dall'invasione di task che dovessero (per qualunque ragione) tentare di agire in zone di memoria "vietate".

Nell'Amiga 2500 UX la scheda A2620 svolge un ruolo fondamentale per tre motivi:

- 1) occorre necessariamente il Motorola 68020 perché la versione System V dello Unix pensata per l'Amiga possa funzionare
- 2) il clock da 14 MHz avvicina le prestazioni della macchina in termini di velocità a quelle di moltissime altre workstation Unix
- 3) la MMU garantisce la protezione indispensabile al sistema operativo Unix.

L'hard disk da 80 MB e lo streamer

Oltre alla scheda acceleratrice già esaminata, dal punto di vista

hardware l'Amiga 2500 UX aggiunge uno streamer da 150 MB, un hard disk da 80 MB, nella cui partizione Unix è presente anche il sistema operativo, e la relativa scheda controller A2090A. La Commodore intende aggiungere ancora, prima di avviare la commercializzazione, una scheda con diverse porte seriali, indispensabile per collegare la macchina in rete con altri terminali e creare quindi un sistema multiutente. Attualmente la presenza di questa scheda non è certa, ma noi ci auguriamo che verrà fatto anche questo sforzo, perché ci sembra che un sistema Unix chiuso nei confronti dei collegamenti in rete scoraggerebbe molti potenziali acquirenti.

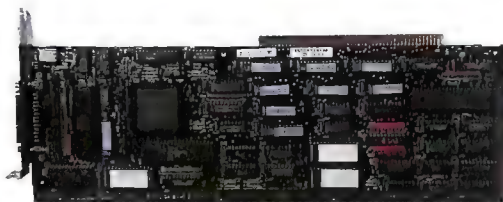
Lo Unix fa uso della memoria virtuale per sopperire ai limiti della RAM disponibile, e per questo l'hard disk è uno strumento fondamentale. Ricordiamo che la "memoria virtuale" è un meccanismo di gestione della memoria che consente ai programmi di "vedere" più byte di RAM di quelli fisicamente presenti nella macchina. Il meccanismo della memoria virtuale fa sì che parte della memoria necessaria al programma risieda nella memoria di massa (l'hard disk), e che venga richiamata solo nel momento di reale necessità, scambiandola ovviamente con altre aree non più necessarie. Tutto è trasparente ai programmi in esecuzione, ed è il sistema operativo ad assicurare questa trasparenza, grazie all'uso della già citata MMU. In parole povere, questo chip è un controllore del bus degli indirizzi; il suo compito è tenere sotto controllo le aree di memoria di cui stanno facendo uso i vari task. Se un task cerca d'indirizzare uno spazio di memoria al quale non può accedere perché è riservato al sistema operativo o a un altro task, oppure se, come accade con la memoria virtuale, il task sta cercando di accedere a una parte di memoria non disponibile perché precedentemente scaricata sul disco, la MMU "avvisa" il microprocessore e quest'ultimo prende le decisioni

opportune. A seconda dei casi, per esempio, blocca il tentativo d'intrusione oppure effettua il caricamento dell'area di memoria richiesta. Si comprende dunque che il tempo d'accesso dell'hard disk influenza pesantemente le prestazioni del sistema a memoria virtuale, visto che questi caricamenti e registrazioni di blocchi di byte possono aver luogo anche a intervalli brevissimi.

L'hard disk che abbiamo visto installato nell'Amiga 2500 UX era un Quantum da 80 MB e tempo di accesso medio pari a 18 ms, suddiviso in due partizioni, una Unix e una AmigaDOS. La capacità di questa periferica è certamente adeguata alle forti richieste di memoria di massa del sistema Unix, ma è il tempo d'accesso medio così ridotto che ci ha fatto definire "superbo" questo hard disk: si tratta di una prestazione eccezionale per un disco installato in un personal computer, niente a che vedere con l'A590 in prova su questo stesso numero, e all'altezza delle prestazioni che deve offrire una buona workstation Unix. La velocità del disco è tale che anche le operazioni svolte in ambiente AmigaDOS risultano rapidissime.

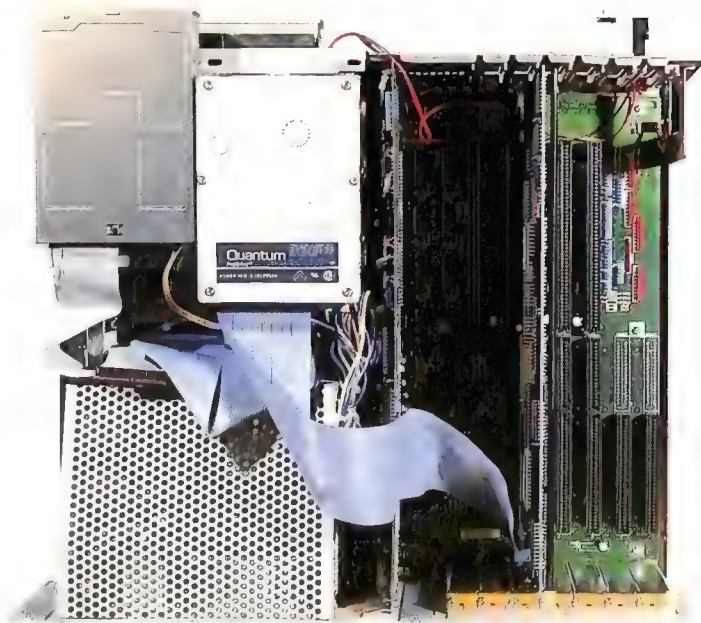
Il collegamento dell'hard disk con il controller A2090A avviene

attraverso una porta SCSI, come per l'A590, e il driver software di controllo è anche in questo caso un dispositivo shared di nome scsi.device (analogo ai dispositivi Parallel o TrackDisk). La scheda



La scheda controller A2090A che gestisce l'hard disk da 80 MB

controller A2090A è inserita in uno degli slot presenti sulla motherboard dell'Amiga. Questo controller, che viene anche venduto separatamente (si veda l'arti-



Si notano l'hard disk Quantum, l'A2620 nel CPU slot, e il controller A2090A sulla destra

colo "Gli hard disk per la famiglia Amiga" pubblicato nel numero 2/88 di *Commodore Gazette*), permette di suddividere un hard disk in partizioni AmigaDOS e non, è in grado di pilotare fino a sette

dispositivi SCSI e due dispositivi ST506, e contiene due ROM che consentono al *Kickstart 1.3* il boot della macchina da hard disk (questa è la maggiore novità rispetto alla precedente versione A2090).

Fisicamente si tratta di una complessa scheda che occupa l'intera profondità dell'Amiga 2000.

Degli 80 MB messi a disposizione dall'hard disk, la partizione attivata sul computer in esame assegnava i tre quarti allo Unix e i restanti 20 MB all'AmigaDOS, una scelta ragionevole, viste le rispettive esigenze di memoria.

A parte qualche anomalia nel funzionamento dei led relativi all'alimentazione e all'attività dell'hard disk, il disco si è comportato molto bene anche in complesse prove di funzionamento svolte in ambiente AmigaDOS. Purtroppo non è assolutamente possibile lo scambio d'informazioni fra l'ambiente Unix e l'ambiente AmigaDOS, e neanche l'accesso alle rispettive partizioni dell'hard disk; questo significa che non ci sono mezzi per importare ed esportare file tra i mondi Unix e AmigaDOS, se non ricorrendo a floppy disk o a un'interfaccia seriale. È una limitazione abbastanza grave, anche se potrà senz'altro essere superata in futuro.

Prima di portare finalmente il puntatore sull'opzione Amiga UX

continua a pagina 56

LA STAMPANTE A COLORI MPS 1224C A 24 AGHI

La MPS 1224C è una stampante Mannesmann Tally (la MT 222) proposta con il marchio Commodore. Questa periferica non promette caratteristiche fantascientifiche, ma riesce a cavarsela bene in quasi tutte le occasioni e molti utenti Commodore ne saranno senz'altro conquistati... e non solo loro.

È una stampante grafica a colori a 24 aghi e 136 caratteri per riga, con velocità massima di stampa pari a 220 cps in modo Draft bidirezionale, che scende a 72 cps in modo Letter Quality. Offre un'interfaccia parallela Centronics e anche un'interfaccia seriale, così da poter essere collegata a qualsiasi computer senza necessità di ulteriori acquisti; peccato solo che il connettore seriale non sia il consueto DB 25 standard, ma un DIN circolare, senz'altro meno usato per questo genere di collegamenti.

Per quanto riguarda l'aspetto esteriore la prima impressione è di sobrietà e pulizia, e soprattutto di notevole robustezza (una caratteristica di cui raramente le stampanti Commodore possono vantarsi). La console di comando, sulla destra del pannello che permette l'accesso alla meccanica di stampa, offre tasti a membrana; questo tipo di tasti, ormai in voga nelle stampanti più moderne, presenta il vantaggio d'isolare completamente i microinterruttori dalla polvere, garantendo loro una vita più lunga. Gli svantaggi sono invece una risposta non ben definita alla pressione e il più rapido deterioramento della membrana rispetto alla plastica. Comunque la console della MPS 1224C si rivela più precisa e sensibile di altre. Tramite i tasti dispo-

gnibili e i led di segnalazione si possono compiere le consuete operazioni di gestione della stampa, ma è anche possibile cambiare la fonte-carattere, il numero di caratteri per pollice (selezionabile fra 10, 12, 15, 17, 20), e scegliere il modo Draft o il Letter Quality. Trattandosi delle operazioni più frequenti, risulta molto comodo poterle effettuare direttamente da console. Inoltre, sempre da console è possibile attivare il menu che permette di alterare permanentemente la configurazione della stampante, senza necessità di agire su inaccessibili DIP switch interni (per fare questo è necessario inserire un foglio di carta, sul quale verrà stampato lo stato dei vari parametri).

Uno sguardo all'interno rivela diversi particolari interessanti sulla meccanica di stampa, sul movimento della testina e sul nastro inchiostrato (che tra le altre cose ha una forma originalissima, e dispone di tre bande colorate e una nera). Il carrello che sostiene la testina e il caricatore dell'inchiostro scorre su un albero orizzontale, che si può regolare tramite una levetta per determinare la distanza tra la testina di stampa e la carta. Il carrello è fissato a un filo d'acciaio collegato al motore passo-passo che provvede a trascinarlo lungo la riga di stampa. Durante i

movimenti del carrello, l'avvolgimento del nastro inchiostrato si ottiene con la rotazione relativa di un ingranaggio lungo una cremagliera fissa disposta parallelamente al percorso compiuto dal carrello lungo la riga di stampa. La scelta di utilizzare per il trascinamento un filo anziché una cinghia, e un solo albero di guida per la traslazione del carrello è una novità nella meccanica di stampa, e una valutazione attendibile diventerà possibile solo con il tempo.

Il funzionamento, sia a colori sia in bianco e nero, appare soddisfacente; esistono i consueti modi di stampa (pica, elite, ecc.), e l'inserimento dei fogli singoli, nonché le operazioni sul rullo e sul trattore, sono pratiche e veloci, nonostante qualche levetta un po' dura. È possibile inserire una cartuccia di fonti aggiuntive e la presenza di un buffer grande fino a 12K snellisce parecchio le operazioni.

Fa piacere riscontrare la presenza di un sistema per "parcheggiare" il modulo continuo senza doverlo estrarre quando occorre stampare su fogli singoli con il trascinamento a frizione; si tratta di una caratteristica importante, che facilita notevolmente lo scambio fra i due trascinamenti. Evidenziamo inoltre il fatto che la MPS 1224C può emulare la EPSON LQ2500, la NEC Pinwriter e la IBM Proprinter XL24.

Una peculiarità di questa stampante è il metodo usato per la stampa a colori: si può infatti scegliere fra due diversi caricatori, uno contenente il nero e i tre colori fondamentali (cyan, magenta e giallo) della sintesi sottrattiva, l'altro contenente il nero e i tre colori fonda-

mentali (rosso, blu e verde) della sintesi additiva. L'utente deve operare sulla configurazione permanente (o via software) per comunicare alla stampante qual è il caricatore installato. L'impiego di una tecnica o dell'altra dipende dalle dominanti di colore presenti nell'immagine e dal tipo di risultato che si desidera ottenere. Va notato infine che la stampa a colori può avvenire anche per sovrapposizione dei colori fondamentali sui singoli punti di stampa, aumentando notevolmente la densità dei colori (in questo caso un'analisi ravvicinata dello stampato non mostra un mosaico di colori fondamentali, come avviene nella stampa tradizionale).

Per quanto riguarda il prezzo, la cifra ventilata (lire 1.200.000, Iva esclusa) ci sembra senz'altro adeguata alle caratteristiche della macchina e in linea con i prezzi delle sue concorrenti.

Per ulteriori informazioni contattare direttamente:

Commodore Italiana spa

(MPS 1224C, L. 1.200.000 Iva esclusa)

Via F.lli Gracchi, 48

20092 Cinisello Balsamo (MI)

(Tel. 02/618321)



del menu iniziale, ci rimane da descrivere lo streamer, il mezzo universalmente adottato per lo scambio di software in ambiente Unix. Nell'Amiga che avevamo in prova, questa periferica era un'unità da mezzo pollice di fabbricazione Wangtek, con una capacità di 150 MB; non abbiamo avuto modo di provarlo, ma non ci aspettiamo spiacevoli sorprese dato che nel mondo Unix si tratta di una periferica usatissima, al pari dei disk drive nel mondo dei PC.

Entriamo in Unix

Entrando in ambiente Unix, cosa che tra l'altro si può fare anche dal *Workbench* mandando in esecuzione un piccolo programma, si perdono tutte le caratteristiche tipiche dell'Amiga e ci si trasferisce in un nuovo mondo: appare infatti uno sfondo color senape sul quale inconsueti caratteri blu ci forniscono notizie sullo stato del processo di attivazione. Lo Unix ha infatti bisogno di svolgere una procedura piuttosto elaborata quando viene attivato e disattivato, ben diversa dalle semplici routine di reset del C-64 e dai file di startup. Non ci inoltreremo nella descrizione di operazioni come il logon oppure lo shutdown; diciamo solo che queste procedure sofisticate servono per evitare conflitti tra i file dei vari utenti e/o task al lavoro nel sistema, e a fare in modo che il sistema venga disattivato soltanto quando tutti sono stati avvertiti e hanno smesso d'impartire comandi.

Si tratta comunque di dettagli ben noti a chi conosce questo sistema operativo, e qui vogliamo mettere in evidenza più le differenze che le affinità con lo Unix standard. Si tratta di uno Unix System V, release 3 AT&T, che

non ha subito quasi nessuna modifica nell'installazione sulla macchina Commodore. L'unica aggiunta di rilievo è un Window Manager, ovvero la possibilità di avere delle "viste" sulle console virtuali (sdoppiamenti o cloni della postazione di lavoro). Intuition non c'entra più nulla, anche perché lo Unix si appropria interamente del sistema e l'unica cosa che rimane attiva è il reset da tastiera.

In pratica ci viene messa a disposizione una console principale più altre quattro console virtuali, attivabili e commutabili attraverso l'uso dei tasti funzione

Le console virtuali sono dotate di un'ombra che muta di spessore con molta precisione a seconda del numero di finestre sovrapposte (e quindi della loro apparente "distanza"), con un risultato molto elegante sul piano estetico.

La main console, quella che viene creata dopo l'attivazione del sistema, non può disporre di finestre (per ragioni di compatibilità). Le console virtuali si trovano infatti all'interno di un loro schermo, alternabile con lo schermo della main; in ogni istante può essere attiva solo una console, che si può riconoscere dal cursore non lampeggiante, al contrario dei cursori nelle finestre delle console non attive. Il refresh delle parti coperte è automatico, e non sappiamo se sia possibile ridirigere l'output di un programma Unix verso un'altra finestra. È invece scontato che esisteranno gli strumenti di programmazione in C per creare task che aprano e gestiscano finestre, ed è probabile, per ragioni di compatibilità dei sorgenti, che tali finestre debbano essere viste dai task come dispositivi di I/O.

Nel Window Manager che abbiamo provato esisteva un bug, in realtà: in certe situazioni non veniva gestito lo schermo a 256 linee del sistema PAL e nella parte inferiore rimanevano visibili alcuni pixel. Ci hanno comunque assicurato che questo problema verrà risolto prima che il prototipo diventi una realtà commerciale.

Abbiamo poi avuto notizia che lo Unix potrà leggere i dischi in formato AmigaDOS, però sembra che nessuno alla Commodore Italiana abbia ancora fatto questo esperimento (forse a causa della difficoltà di predisporre driver di accesso a disk drive da 3,5"). Per il resto, tenendo conto del limitato numero di prove che sono state effettuate, possiamo dire che il sistema si comporta egregiamente.



Le console virtuali a finestre gestite dal Window Manager

e del mouse. Il mouse consente d'interagire con queste console virtuali e di variarne il numero in un modo che ricorda più i computer Xerox che l'interfaccia utente dell'Amiga.

Premendo il pulsante destro del mouse appare accanto al puntatore un riquadro rosso composto da icone; selezionandole con il mouse, si possono compiere diverse operazioni sulle finestre che contengono le console virtuali... dimensionamento, spostamento, alterazione dei reciproci livelli di profondità, scelta della finestra attiva e così via. Il metodo è abbastanza immediato, anche se non quanto i gadget di Intuition.

te: si lavora in maniera efficiente, e tutto sembra seguire il classico standard Unix.

Ci auguriamo ora che questa interessante proposta venga arricchita da adeguati pacchetti applicativi prima di essere messa in commercio, e che le molte promesse vengano mantenute. Per la Commodore, l'Amiga 2500 UX significa un'importante sfida a un mercato certamente non facile, oltre che una grossa novità per i suoi utenti.

Conclusioni

Abbiamo riservato parecchie considerazioni per quest'ultima parte dell'articolo. In una successiva e più approfondita prova, che si spera potrà avvenire sulla versione definitiva dell'Amiga 2500 UX, avremo modo di verificare le anticipazioni illustrate e di scendere maggiormente nei dettagli. Per ora possiamo affermare con certezza che il sistema funziona, che il prototipo esaminato verrà ulteriormente migliorato, e che se le promesse saranno mantenute non mancherà certamente di stimolare il mercato. L'unica anticipazione che non siamo riusciti a strappare riguarda il prezzo dell'A2500 UX, sul quale la Commodore non ha voluto sbilanciarsi prima del tempo. Evidentemente l'analisi dei costi non può essere effettuata finché non viene decisa esattamente la configurazione definitiva. Per esempio, non è esclusa la commercializzazione di un kit che consenta di trasformare il proprio Amiga 2000 in una workstation Unix come quella che abbiamo appena descritto, ma è una decisione che si scontrerebbe con l'intenzione di vendere una macchina applicativa chiavi in mano e quindi c'è da aspettarsi che venga scartata.

Sul prezzo dell'Amiga 2500 UX possiamo fare solo una considerazione basata sulla nostra esperienza: è probabile che agli occhi dei potenziali acquirenti dell'Amiga 2000 apparirà un prezzo elevato, mentre agli occhi di chi conosce il

mercato delle workstation Unix apparirà decisamente vantaggioso. È certo che la Commodore punterà sull'economicità del prodotto per far leva sul mercato, ma l'hardware non verrà comunque regalato. Se l'Amiga 2000 ha un prezzo accessibile è anche perché viene commercializzato privo di hard disk, con un solo megabyte di RAM e senza coprocessore matematico... Se all'Amiga 2000 (L. 2.715.000 + Iva) iniziamo ad aggiungere una scheda A2620 (L. 2.700.000 + Iva) e una scheda controller A2090A (L. 1.900.000 + Iva nella versione con hard disk A2094 da 40 MB) entriamo evidentemente in una fascia di prezzi nettamente più elevati di quelli ai quali la serie Amiga ci aveva abituati; se poi a questo hardware di base andiamo ad aggiungere la differenza di costo di un hard disk da 80 MB con tempo d'accesso medio di 18 ms, di uno streamer da 150 MB, del sistema operativo Unix, della scheda che fornirà le varie porte seriali e del monitor, si intuisce facilmente che quando l'Amiga 2500 UX entrerà nel catalogo ufficiale avrà un prezzo certamente considerevole (ma che comunque dovrebbe essere più che vantaggioso rispetto agli elevatissimi costi di una workstation Unix).

Se questa macchina verrà adeguatamente assistita, sarà proprio il suo prezzo vantaggioso a catalizzare l'interesse di moltissimi professionisti, e perché no, anche di università ed enti di ricerca. Certo, non è facile entrare in questi settori, ma neanche impossibile, purché la macchina venga presentata al pubblico con il necessario corredo di applicazioni e commercializzata tramite punti di vendita seri, qualificati e competenti.

Viceversa, si può facilmente prevedere che nessuno di coloro che hanno comprato l'Amiga perché costa poco e offre molto (anche sul piano del divertimento) affronterà una spesa del genere soltanto per procurarsi un ingegnoso passatempo. Tutti coloro che con il computer deside-

rano semplicemente divertirsi troveranno probabilmente il mondo Unix un po' troppo tetro e oneroso per inoltrarcisi.

Per quanto riguarda l'A2620, che nel catalogo Commodore è presente anche come prodotto completamente indipendente dall'Amiga 2500 UX, possiamo affermare che si tratta di una scheda acceleratrice così sofisticata e completa da riuscire a trasformare l'Amiga 2000 radicalmente. Il discorso per quel che riguarda la fascia di utenza è quindi senza dubbio aperto a tutti gli hobbisti evoluti e a chi usa la macchina professionalmente in ambiente AmigaDOS.

Con i suoi 2 MB di RAM a 32 bit espandibili fino a quattro, un clock da 14 MHz, il coprocessore matematico 68881, e soprattutto con il Motorola 68020, questa scheda permette all'Amiga di raggiungere prestazioni che giustificano largamente il suo prezzo. In campo grafico, nella generazione d'immagini tramite la tecnica ray tracing, nell'uso dell'Amiga per creare animazioni e slide-show, nell'impiego di CAD, nelle simulazioni di modelli matematici, e in tutti quei settori d'impiego dell'Amiga che richiedono velocità e potenza di calcolo, la scheda A2620 riscuoterà sicuramente un vasto successo.

Le software house che producono pacchetti applicativi professionali per l'Amiga stanno già creando nuove versioni dei loro programmi che sfruttano questa scheda, e questo, oltre a costituire un importante segno di fiducia verso il prodotto, è il primo indispensabile passo verso una larga diffusione.

*Per ulteriori informazioni
contattare direttamente:*

Commodore Italiana spa
(A2500 UX prezzo da stabilirsi,
A2090A + A2094 L. 1.900.000,
A2620 L. 2.700.000 Iva esclusa)
Via F.lli Gracchi, 48
20092 Cinisello Balsamo (MI)
(Tel. 02/618321)

A-MAX, UNA MELA PER L'AMIGA

La ReadySoft ha appena lanciato sul mercato statunitense il pacchetto *A-MAX*, il primo emulatore Macintosh per l'Amiga. Si tratta di una combinazione software/hardware che per il modico prezzo di 199,95 dollari trasforma l'Amiga in un Apple Macintosh, permettendo così agli utenti Commodore di accedere alla vasta biblioteca di applicazioni che ha fatto la fortuna del glorioso Mac. Ma che cosa significa "emulare" un'altra macchina? E com'è possibile riuscirci? Cerchiamo di rispondere brevemente a queste domande prima di presentare tutte le notizie sull'*A-MAX* di cui disponiamo al momento.

Un emulatore ha il compito di far apparire un computer, nel suo insieme, *come se fosse* un altro. Le emulazioni si differenziano in due livelli fondamentali. Se i due computer possiedono CPU diverse, come per esempio il C-64 e l'Amiga, l'emulatore deve anzitutto far sì che il microprocessore di cui si dispone possa eseguire i codici macchina dell'altra CPU, quella del computer da emulare; dal momento che non si può alterare il set d'istruzioni di una particolare CPU, quello che si fa di solito è decodificare istruzione per istruzione, e in tempo reale, i codici macchina previsti dalla CPU da emulare. Si tratta di un'operazione molto complessa, e costosa in termini di velocità d'esecuzione, in quanto occorre praticamente riprodurre via software la logica hardware di decodifica presente all'interno della CPU da emulare. Se invece le due macchine possiedono la stessa CPU, come appunto il Macintosh e l'Amiga, lo sforzo di decodifica illustrato non è più necessario, e le prestazioni dell'emulatore migliorano di conseguenza.

Restano però da riprodurre - con l'hardware di cui si dispone o tramite opportune espansioni - le caratteristiche di un hardware (CPU a parte) completamente diverso. Cioè si deve fare in modo che le funzioni svolte dai chip del computer da emulare vengano imitate nella maniera più fedele possibile. Purtroppo, anche questo secondo livello di emulazione, apparentemente più banale, presenta problemi tali che spesso le prestazioni dell'emulatore raggiungono a malapena la sufficienza.

Se i due livelli di emulazione sono stati realizzati alla perfezione, non occorre far altro che installare nel nostro computer, sotto il controllo dell'emulatore, il sistema operativo del computer da emulare e tutto funzionerà regolarmente. Ma che non occorra nessuna modifica del sistema operativo è un'eventualità molto rara: l'emulazione dell'hardware, infatti, non è quasi mai completamente affidabile e in genere occorre ritoccare, più o meno pesantemente, il sistema operativo del computer da emulare.

Alla luce di questo discorso, analizziamo brevemente quanto è stato fatto finora per cercare di trasformare l'Amiga in un altro computer. Tutti ricordiamo il *Transformer* (il primo emulatore MS-

DOS per l'Amiga), le schede Bridgeboard XT e AT, e i due emulatori C-64: in ognuno di questi casi è stato reso possibile l'impiego sull'Amiga di programmi scritti per altri computer. Ogni volta i problemi sono stati affrontati in maniera diversa: le due schede Bridgeboard, per esempio, non sono veri emulatori. Contengono infatti la riproduzione completa dell'hardware e del firmware che costituiscono un PC MS-DOS compatibile dotato di 8086 o 80286. La loro filosofia è che per emulare un computer con il massimo grado di fedeltà, l'unica via è inserire il computer stesso all'interno di quello che lo deve emulare... una via non sempre praticabile, soprattutto per ragioni economiche. Il *Transformer*, invece, è un vero emulatore MS-DOS, cioè un sofisticato programma che interpreta i codici macchina ed emula l'hardware; trattandosi però di un emulatore esclusivamente software presenta gravi limiti di compatibilità e di velocità. Gli emulatori C-64, invece, riescono a ottenere buoni risultati sotto ogni profilo, soprattutto perché la macchina da emulare è più lenta e più semplice dell'Amiga.

Questo ci permette di trarre una conclusione di valore generale: quando si cerca di emulare un computer dalle capacità analoghe o addirittura superiori alla macchina di cui si dispone, il compito si dimostra molto arduo; la situazione è invece più rosea quando si va nella direzione opposta. Per esempio, l'emulazione software dell'MS-DOS disponibile per il computer Archimedes della Acorn riesce a svolgere lo stesso lavoro di decodifica del *Transformer* garantendo però una velocità una volta e mezza superiore a quella dell'IBM XT, grazie alla potenza enormemente superiore della sua CPU RISC.

Il Mac in ambiente Commodore

L'*A-MAX* è un emulatore nel senso stretto del termine: il software agisce in maniera tale da appropriarsi integralmente dell'Amiga e ricreare l'ambiente hardware del Mac anche per quanto riguarda l'accesso ai disk drive. È accompagnato da un modulo d'espansione hardware destinato a contenere le Eprom di sistema del Macintosh, sia quelle da 64K del primo Mac sia quelle da 128K del Mac Plus. Inoltre viene messa a disposizione una porta SCSI per collegare un disk drive Apple da 800K (sulla cui utilità discuteremo in seguito) e una porta per il collegamento di un altro disk drive Amiga. Un aspetto interessante del modulo d'espansione è che viene collegato alla porta disk drive dell'Amiga, al fine di assicurare la compatibilità con tutti i modelli della linea Amiga, dal momento che il connettore per disk drive è sempre lo stesso. È interessante anche notare che le ROM contenenti le routine di base del Macintosh non vengono fornite con il prodotto, a causa di problemi di

copyright: occorre quindi procurarselo di persona.

Com'è noto il Mac, al pari dell'Amiga, possiede un Motorola 68000, e questo ha facilitato notevolmente la creazione dell'emulatore A-MAX. Infatti non c'è stato bisogno di creare un parser che decodificasse i codici macchina dei programmi per il Mac. Per di più va detto che usando le ROM del Mac Plus è possibile utilizzare a piena potenza una CPU 68020 eventualmente installata nell'Amiga.

La grande flessibilità dei chip grafici dell'Amiga ha permesso di ricreare senza troppa fatica la risoluzione video standard del Macintosh di 512 x 342 pixel in interlace. È stato invece necessario riconfigurare la memoria RAM in quanto nel Mac è organizzata in un unico blocco a partire dall'indirizzo zero, mentre nell'Amiga è ripartita in diverse aree del campo indirizzabile non contigue fra loro. La questione è stata risolta in maniera molto efficace e anche i programmi che superano la barriera dei 512K vengono eseguiti senza difficoltà (per i primi 512K il problema non si pone perché la memoria chip è continua per definizione e sistemata a partire dall'indirizzo zero).

Con i disk drive si sono incontrati problemi molto maggiori. I formati adottati dalle due macchine sono infatti abbastanza diversi, sia dal punto di vista hardware sia da quello software: pur adottando lo stesso sistema MFM per la registrazione dei dati sul supporto, nei disk drive Apple la velocità di rotazione varia a seconda della traccia alla quale si accede. La meccanica dei disk drive dell'Amiga può replicare solo due delle cinque velocità del Mac, e quindi i dischi formattati dal Macintosh non possono essere letti direttamente dall'Amiga. Si deve per forza ricorrere a un'opportuna utility (da usare con il Mac) che ripartisce i vari blocchi in tre dischi leggibili dall'Amiga. L'A-MAX provvederà poi a riunire il tutto in un unico disco in formato Amiga. Comunque, la soluzione migliore per importare file e programmi da dischi originali Macintosh rimane un disk drive Apple originale collegato alla porta SCSI del modulo hardware che accompagna l'emulatore A-MAX.

A proposito dei driver software per le unità a disco, c'è da dire che a complicare ulteriormente le cose è venuta l'impossibilità di utilizzare le routine interne dell'Amiga (quelle contenute nel dispositivo TrackDisk) in quanto le operazioni sarebbero diventate troppo lente. Sono quindi state riscritte da capo anche le routine di livello più basso, quelle a contatto diretto con l'hardware.

Sembra che le altre periferiche, invece, non abbiano subito traumi rilevanti nella trasformazione. Le porte seriale e parallela dell'Amiga possono essere utilizzate tranquillamente dal sistema operativo del Macintosh, con in più la possibilità di sceglierne le modalità d'uso. Infatti il Mac dispone di due porte seriali e nessuna parallela, e quindi sta a noi decidere come le porte del computer Commodore dovranno corrispondere a quelle originali.

Le capacità grafiche e i colori dell'Amiga non vengono utilizzati quasi per niente, in quanto l'emulatore trasforma l'Amiga in un Mac della prima serie,

per il quale non era prevista la grafica a colori (lo è stata solo a partire dal Macintosh II); possiamo però scegliere con quale coppia di colori lavorare.

Lo stesso discorso vale per i suoni, che si limitano a un semplice "beep", l'unico suono che viene utilizzato dalla maggior parte dei programmi professionali per il Mac. I giochi fanno uso di una speciale tecnica, chiamata PWM (modulazione della durata d'impulso), tramite la quale riescono a ricavare suoni più sofisticati, in genere provenienti da digitalizzazioni, intervenendo direttamente sull'hardware. Almeno in questa versione dell'A-MAX tale tecnica non è utilizzabile con l'Amiga (sembra a causa di problemi con il DMA).

Infine va segnalato che attualmente non viene riconosciuto nessun hard disk, ma per il futuro sono previste modifiche che renderanno possibile il collegamento di alcuni hard disk dedicati al Mac e anche l'accesso a una partizione di un hard disk per l'Amiga.

Concludiamo parlando dei programmi per il Macintosh che sembrano non aver problemi con l'A-MAX. Oltre a *MultiFinder*, sono state provate moltissime applicazioni professionali, tra cui *HyperCard*, *Excel*, *PageMaker*, *Word*, *MacPaint* (tutte le versioni), *MacWrite*, *FullPaint*, *MacDraw II*, *LightSpeed Pascal*, *LightSpeed C*, *SuperPaint* e *TMON*. Inoltre funzionano perfettamente anche tutte le versioni del sistema operativo fino alla 6.00. Tra i programmi con difficoltà più o meno gravi troviamo invece i giochi, che funzionano solo in minima parte, alcuni programmi di comunicazione (tranne *MacTerminal* che funziona perfettamente), e alcuni programmi che consentono d'interfaciare il Mac con tastiere MIDI. Tutti questi tipi di programma, infatti, fanno parecchio affidamento sull'hardware ed è ovvio che incontrino delle difficoltà quando l'hardware viene a mancare.

La velocità che si ottiene attraverso l'A-MAX è la stessa del Macintosh Plus originale, in particolare per quanto riguarda l'accesso ai dischi. Considerando che il prezzo di questo pacchetto è di soli 199,95 dollari, contro il costo quasi dieci volte superiore della macchina Apple, è facilmente prevedibile un buon successo per l'A-MAX: poter utilizzare le più famose applicazioni professionali per il Mac (con le quali la compatibilità sembra accertata) sarà una grossa tentazione per parecchi utenti. Infine, è lecito aspettarsi alcune migliorie nelle prossime versioni dell'A-MAX se, come afferma il suo autore, l'intero progetto è nato soltanto per vedere come se la sarebbe cavata l'Amiga in un compito che l'Atari ST aveva già affrontato.

**Per ulteriori informazioni
contattare direttamente:**

ReadySoft
(A-MAX, \$199,95)
P.O. Box 1222
Lewiston, NY 14092, USA
(Tel. 001/416-7314175)

LA VIOLENZA CORRE SUL VIDEO

Forse i videogiochi nazisti sono soltanto un abbaglio di giornalisti in cerca di sensazionalismo, ma nella produzione attuale c'è anche di peggio. Un'inchiesta di Commodore Gazette nel mondo del software

di Fabio Rossi

È una serata dell'inverno scorso: come di consueto, nelle loro case milioni d'italiani si siedono davanti al televisore, sintonizzati sulla prima rete nazionale. La trasmissione è *TG1 Sette*, il settimanale d'informazione della Rai.

Fra i servizi proposti, ce n'è uno che colpisce le famiglie molto più delle guerre in Medio Oriente o dell'assurda serie d'incidenti o aerei che hanno provocato in poche settimane centinaia di vittime. L'argomento è la diffusione del pericoloso fenomeno dei videogiochi nazisti. Oggetto anche di parecchi articoli sulla stampa italiana specializzata e non, i "videonazi" hanno titoli raccapriccianti: *Hitler Dictator*, *Antitürkesh Test*, *Treblinka*, *Smash the Jews*. Grazie a *TG1 Sette*, ora sconvolgono anche dal video centinaia di famiglie che non avevano il minimo sentore del fenomeno. Per completare il quadro, il servizio è accompagnato da un'intervista che calca la mano sul pericolo di diseducazione a cui sono sottoposti i giovanissimi fruitori dei videogiochi.

Dopo un'introduzione adeguatamente allarmistica, vengono infatti poste alcune domande all'autore di *Hitler Dictator* (ariano, naturalmente). Le sue risposte,

come afferma un lettore della rivista specializzata *The Games Machine*, sono "da far accapponare la pelle", e alla fine del servizio sono in molti a credere che anche *Space Invaders* abbia implicazioni antisemitiche.

nare. Persa subito la speranza di trovare questo genere di prodotti sul mercato ufficiale, abbiamo indagato fra le maggiori associazioni di pirati europei, ma ancora con risultati scoraggianti.

Più li cercavamo e più i video-



Una schermata di *Stalag*, contrabbandato come nazigame

L'inesistente invasione

Vista la preoccupazione generale sollevata dai numerosi articoli che sulla carta stampata hanno fatto da cornice al servizio televisivo, abbiamo voluto indagare anche noi sul fenomeno dei "nazigame", per toccare con mano i famigerati dischi che decine di giornalisti non specializzati avevano evidentemente potuto esami-

nazi sembravano come il leggendario tesoro delle SS: tutti sanno che esiste, ma nessuno riesce a trovarlo. Proprio quando avevamo perso ogni speranza di rintracciare anche solo un programma per documentare fotograficamente l'articolo, la vasta rete di contatti intrecciata durante le ricerche ha dato i suoi frutti: finalmente ci siamo visti recapitare un disco per C-64, con un'inte-

stazione (scritta a penna) che non lasciava dubbi. Avevamo *Smash The Jews!*

Schiaccia gli ebrei

Una volta caricato in memoria, *Smash the Jews* si è rivelato un vecchio giochino del 1984, originariamente intitolato *Stalag*, cui era stata aggiunta una schermata introduttiva con il nuovo titolo, e nient'altro.

Stalag (ci rifiutiamo di usare il nome della versione piratata) è un programma di una semplicità disarmante: da un gruppo di baracche situate al centro dello schermo, minuscoli omini realizzati con un singolo carattere grafico rilocato scavano alcuni tunnel verso i bordi dell'area di gioco, lasciandosi alle spalle rozze strisce bianche. Il giocatore, nei panni di un piccolo sprite che si muove mimando il passo dell'oca, deve intercettare i prigionieri. Con il passare del tempo i fuggitivi diventano sempre più veloci e numerosi, sino all'inevitabile momento del game over, che sancisce inesorabilmente la vittoria dei prigionieri. A seconda del livello di difficoltà, la sentinella viene ostacolata dalla presenza di cani da guardia che, lungi dal compiere il loro dovere, preferiscono scorrazzare per il cortile del campo attendendo alle caviglie del sorvegliante. Tutto qui.

A *Stalag* è stata semplicemente aggiunta la nuova schermata farneticante (realizzata da un pirata inglese, a quanto sembra) e la difficoltà di reperirlo è indice dello scarso interesse che il programma (vecchia o nuova versione) ha suscitato fra gli utenti.

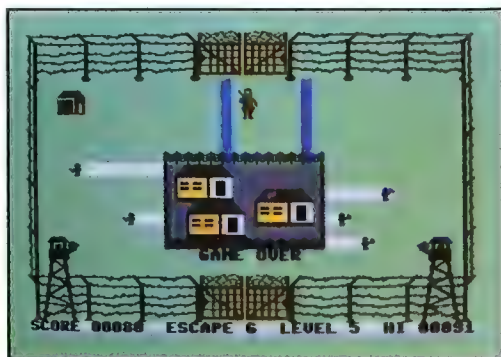
Ora che abbiamo riportato il fenomeno del software "nazista" a una dimensione più contenuta ma più realistica, rimarrebbe da considerare lo *Stalag* originale. Per quanto le interpretazioni possano essere discordi, la nostra impressione è che *Stalag* non sia altro che una mediocre ricostruzione di un episodio della Seconda guerra mondiale. I prigionieri

(non ebrei ma americani, come dimostra il jingle che accompagna la loro fuga) annullano con la loro superiorità numerica il dominio della guardia, peraltro ridicola nella sua incapacità di controllare persino i suoi stessi cani.

Alla fine il gioco si risolve sempre con un'evasione in massa dal campo. In *Stalag* non c'è ombra di violenza: la sorte dei prigionieri intercettati è lasciata all'immaginazione del giocatore, e la logica stessa del gioco suggerisce una nuova segregazione indo-

non è altro che una mistificazione, ci fanno ritenere che tutto questo parlare di nazismo non sia altro che un abbaglio della stampa non specializzata alla ricerca di titoli a sensazione. Del resto nessuno si è mai sognato di alzare l'indice accusatore su telefilm come *Gli eroi di Hogan*, una serie televisiva la cui trama è identica a quella di *Stalag*.

Resta invece l'inquietante realtà del software "ultravioletto". Da sempre un certo genere di videogiochi si basa sull'elimina-



Sopra: i prigionieri che fuggono nella schermata di *Stalag*. Sotto: *Operation Wolf*, un gioco violento per C-64 e Amiga

lore, che permetterà immediatamente un nuovo tentativo di fuga.

La vera ultraviolenza

Il fatto di non aver potuto reperire software nazionalsocialista, né presso i negozi né presso i pirati, e l'aver scoperto che *Stalag*

zione totale dell'avversario, con infinite variazioni sul tema. L'eliminazione può riguardare il personaggio controllato da un altro giocatore o una flotta di astronavi aliene, ma nei primi videogame era sempre assolutamente asettica. Era un atto violento come abbattere i birilli in una partita di bowling.

Poi, a mano a mano che le

ALLA RICERCA DEL NAZISTA PERDUTO

Quando abbiamo tentato di procurarci i programmi accusati di "videonazismo" che hanno scatenato negli ultimi tempi gli strali della stampa italiana, abbiamo scoperto una realtà molto diversa da quella descritta da quotidiani e trasmissioni televisive. I "nazigame" sembravano non esistere affatto, e solo la creazione di una fitta rete di contatti in molti Paesi europei (Italia, Inghilterra, Germania, Belgio, Francia, Spagna e Olanda) ci ha permesso di giungere a qualche risultato.

Dopo aver abbandonato quasi immediatamente le ricerche tramite i canali ufficiali (negozi e importatori), le nostre indagini si sono spostate sul circuito pirata. In Italia, due dei maggiori esponenti di questo ambiente dichiarano di non aver mai visto giochi nazisti se non il solito *Stalag* (del 1984) e *Third Reich* (che è la trasposizione elettronica di un noto gioco da tavolo del tutto apolitico). Dalla Spagna e dalla Francia vengono notizie analoghe. Le prime risposte interessanti sono arrivate dagli hacker britannici e olandesi, che hanno affermato di essersi rifiutati d'inserire nelle loro "liste" i giochi estremisti per una specie di autoregolamentazione, mentre in Germania il fenomeno è stato definito come "marginale e passeggero, ormai concluso da molti mesi".

Eccezion fatta per il caso di *Stalag* di cui parliamo dettagliatamente nel corso dell'articolo, l'unico gioco d'ispirazione nazionalsocialista che siamo riusciti a rintracciare è scritto per computer IBM, ha una struttura esclusivamente testuale che ne impedisce la diffusione al di fuori dei Paesi di lingua tedesca ed è relegato a un BBS situato in Belgio. Siamo riusciti ad arrivare a *Dachau General* (questo è il nome del gioco) solo grazie a una complicata serie di collegamenti via modem, rintracciando il programma in questione in una sezione del sistema protetta da diversi livelli di sicurezza, che ne impediscono la visione a chi non sia un vero hacker.

Queste scoperte mettono in evidenza che la stampa italiana si è lasciata andare a una serie di conclusioni affrettate sulla base di qualche confusa nota di agenzia, ingigantendo quello che è un fenomeno pressoché inesistente, relegato in realtà ad ambienti iniziatici e quasi inaccessibili. In realtà, anche se le nostre ricerche lungo i tortuosi canali degli ambienti pirati avessero dato qualche frutto, le conclusioni non sarebbero state troppo diverse. Prima di parlare di "fenomeno di costume" ci vuole ben altro che due o tre programmi impossibili da raggiungere per l'utente medio. Ma gli errori dei giornali non sono certo una novità (ricordate la clamorosa cantonata presa l'anno scorso da *Repubblica* sull'argomento dei virus elettronici?). L'unica differenza è che a questo autodafé hanno preso parte anche personaggi della stampa specializzata: è il caso di Francesco Carlà. Le foto che accompagnano il suo articolo apparso sulle pagine dell'*Espresso* il 2 aprile 1989 arrivano chiaramente a definire "nazigame" il povero *Rocket Ranger* della Cinemaware (in cui l'eroe combatte contro i nazisti), che aveva probabilmente l'unico torto di offrire schermate con svastiche in evidenza, ideali per illustrare l'articolo. **F.R.**

capacità grafiche e sonore degli home computer miglioravano e la memoria disponibile aumentava, i programmatori hanno potuto rendere sempre più realistiche e più cruento queste distruzioni, e oggi è possibile caricare sul proprio computer giochi che offrono bersagli umani da massacrare e violenza gratuita in quantità impressionante. Titoli come *Opera-*

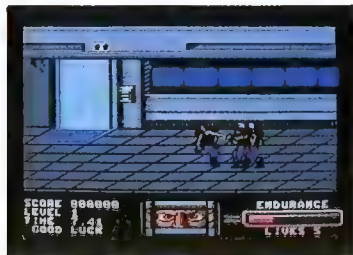
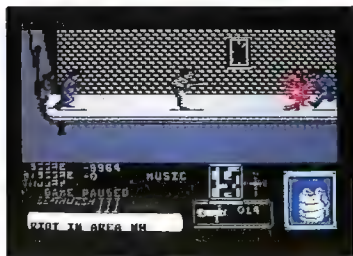


In *Techno Cop* un poliziotto elimina gli avversari con grande spreco di sangue

tion Wolf o *P.O.W.* per l'Amiga o come *Project: Firestart* e le avventure di Rod Pike (*Dracula*, *Jack the Ripper* ecc.) per il Commodore 64, basano il loro successo e le loro impressionanti vendite proprio sulla presentazione di corpi umani smembrati e sanguinanti e sul massacro indiscriminato di avversari disegnati con notevole realismo.

Un elenco dei programmi che

fanno ricorso ad atti di violenza più o meno realistici occuperebbe forse più pagine di quelle che compongono questo numero della rivista, ma un semplice sondaggio e un'occhiata al software più venduto (su tutto il mercato mondiale) ci dicono che i giochi di questo tipo hanno un successo molto superiore a quello che ci si potrebbe aspettare, e proprio



Tre giochi "violenti". Dall'alto: *Infernal Runner*, *Deathwish III* e *Target Renegade*

presso un pubblico di giovanissimi.

Le ragioni di questo orientamento sono semplici e sono le stesse che sociologi e psicologi hanno identificato studiando altre forme d'intrattenimento "violento", come i film splatter, il rock heavy metal o particolari sport come il wrestling: in una società che tende a reprimere l'atavica tendenza degli esseri umani a

VIDEONASTY D.O.C.

Ecco un tentativo di hit parade dei giochi ultraviolenti, che abbiamo diviso fra programmi per home computer e giochi da sala. Per ogni titolo abbiamo indicato fra parentesi le caratteristiche salienti che giustificano la qualifica di "videonasty", sperando che la loro assurdità di fondo sia evidente per tutti i lettori.

Sale giochi:

1. Torture Dungeon (impugnando una pistola a fotocellula si colpiscono impotenti vittime legate a svariati strumenti di tortura, ottenendo in cambio urla digitalizzate e sanguinosi smembramenti).

2. Splatter House (Jason, il protagonista della serie cinematografica *Venerdì 13*, affronta ogni sorta di mostri macellandoli con una sega a catena che ne fa schizzare i liquidi organici sulle pareti, e li trasforma in poltiglie agonizzanti).

3. N.A.R.C. (due agenti di polizia, durante un'operazione antidroga, sparano a personaggi digitalizzati che hanno le fattezze di alcuni noti attori).

4. Operation Thunderbolt (due mitragliatori Uzi vibrano nelle mani dei giocatori, che sparano a immagini molto realistiche senza far troppa distinzione tra guerriglieri e innocenti passanti).

5. Operation Wolf (il precursore del titolo precedente, con una sola mitraglietta e convertito anche per microcomputer).

Home computer:

1. Techno Cop (un poliziotto elimina i suoi avversari con una potentissima pistola che li fa esplodere e li riduce in masse tremanti e sanguinolente, ed è possibile uccidere senza motivo bambini e altri innocenti).

2. Deathwish III (basato sulla serie de *Il giustiziere della notte*, il gioco vede un Charles Bronson digitale armato di bazooka ridurre a brandelli criminali, prostitute e comuni passanti).

3. Le avventure di Rod Pike (durante lo svolgimento dei giochi testuali appaiono improvvisamente alcune realistiche schermate grafiche che rappresentano teste mozzate, corpi sventrati e amenità simili, che vengono riprese dalle descrizioni testuali).

4. Thai Boxing (il gioco consiste in un incontro di boxe thailandese. Il volto dell'avversario, sempre più sanguinante e tumefatto compare in un'apposita finestra grafica).

5. Infernal Runner (gioco di risalita primitivo ma granguignolesco, uscito nel 1984, nel quale il protagonista era trafitto da trappole acuminata, smembrato da mostri, fatto esplodere con abbondante spreco di sangue e così via).

prevalere l'uno sull'altro mediante l'uso della forza, le forme di evasione che si basano su questi aspetti dell'animo umano vengono usate come "valvole di sfogo".

Nei videogame, in più, si possono ottenere soddisfazioni che altri mezzi d'intrattenimento non permettono. In primo luogo la possibilità di essere protagonisti dell'azione, anziché osservatori passivi. In secondo luogo la libertà dalle pastoie delle leggi fisiche, che consente di portare la violenza digitale alle conseguenze più estreme e fantasiose. Entrando in una qualunque sala giochi, oggi è già possibile sventrare con una sega elettrica i propri avversari e ridurli a mucchietti di carne sanguinante (*Splatter House*) o addirittura sfogare i propri istinti sadici sparando con una pistola a fotocellula contro esseri umani prigionieri di strumenti di tortura (*Torture Dungeon*, solo di recente ritirato dal mercato extra-nipponico). Ma le console delle sale giochi hanno capacità spesso molto superiori a quelle degli home computer, e la fantasia dei programmatori fino a oggi si è rivolta soprattutto in quella direzione. C'è da chiedersi quali potranno essere gli sviluppi violenti dei videogame via via che aumenteranno le potenzialità dei computer domestici...

Il futuro dell'omicidio elettronico

Non c'è dubbio che capacità grafiche e sonore sempre più evolute renderanno possibile portare la rappresentazione dell'omicidio a livelli di realismo sempre crescenti, e l'annuncio di nuovi sistemi come la console Konix, che darà supporto a periferiche come fucili a fotocellula dotati di rinculo e visori tridimensionali avvolgenti, rendono plausibili visioni cyberpunk di un futuro in cui non sarà facile distinguere la morte sul video da quella reale.

Già oggi c'è chi si preoccupa dei possibili influssi negativi dei videogame ultraviolenti (che, a

differenza dei videogiochi nazisti, esistono davvero). Parecchi giochi sono stati denunciati alle autorità e sottoposti a censura grazie all'intervento di gruppi di genitori preoccupati, e proprio nella Germania "nazista" è stata proibita la vendita dei programmi che presentano un eccesso di violenza esplicita.

Se in Germania e in generale

a un videogioco, ma è opinione comune che con l'evolversi delle tecnologie e della qualità dei prodotti i rischi aumenteranno sempre più.

Facendo un paragone con gli appassionati dei film horror/splatter, che si trovano bersagliati da immagini simili a quelle di alcuni videonasty, e rammentando il concetto di "valvola di



Una schermata di *Barbarian*, un gioco "vietato ai minori"

nell'Europa del nord esistono leggi specifiche e un codice di autoregolamentazione delle vendite che ha portato per esempio a vietare la vendita ai minori di programmi come *Wolfman* (della serie di Rod Pike) o *Barbarian* della Palace, in cui il sangue scorre a fiumi, in tutto il resto del mondo non c'è nessun controllo sulla diffusione della violenza digitale. Per quanto un intervento di censura sia sempre spiacevole, resta il fatto che la maggior parte delle persone che sono venute a contatto con quel software ultraviolento che gli inglesi hanno definito "videonasty" concorda con l'idea di controllarne in qualche modo le vendite.

La preoccupazione più diffusa è ovviamente che una simile forma di messaggio violento, unita al ruolo attivo del giocatore, provochi nell'utente non preparato (minori, ma anche adulti particolarmente sensibili) shock e disturbi della personalità che potrebbero persino portare a comportamenti pericolosi. A tutt'oggi non si sono registrati casi di violenza ispirata

sfogo", alcuni operatori specializzati (principalmente produttori di software, ma anche giornalisti specializzati) hanno obiettato che queste preoccupazioni potrebbero rivelarsi infondate, e che al contrario un'indigestione di violenza simulata potrebbe limitare il desiderio di violenza reale. Che l'oggetto del dibattito siano film, libri, musica o videogame, in ogni discussione sull'argomento si ritrovano puntualmente queste due posizioni, con esperti di chiara fama che discutono la questione portando a sostegno delle loro tesi studi ed esperimenti.

Chi scrive non può che concludere con un inquietante interrogativo: è proprio vero che a prodotti come i videogiochi può essere attribuita una funzione catartica, o stiamo avvicinandoci al giorno in cui troveremo "normale" che un uomo venga fatto a pezzi da una raffica di mitra sparata in mezzo alla folla? Forse, nella nostra collezione di software abbiamo qualcosa di peggio dei videonasty...

L'ULTRAVIOLENZA: PARLANO GLI OPERATORI DEL SETTORE

Abbiamo chiesto a persone coinvolte a titolo diverso nell'industria del software, di esprimere un'opinione sul fenomeno dei videogiochi nazisti e dell'ultraviolenza.

Le risposte, come si può vedere, hanno manifestato in genere la tendenza a minimizzare

Stefan Roda, tecnico della software house Italvideo: da molto tempo, ormai, il tema conduttore dei videogiochi d'importazione è quello della cosiddetta "ultraviolenza" (come la definiscono con convinzione alcune riviste del settore). Alla diffusione di questo tipo di software hanno contribuito diversi fattori, tra cui alcune ricerche di mercato inglesi (che seguono da vicino l'effetto dei mass-media sulla popolazione giovanile) e una certa mancanza di creatività nella ricerca di temi originali.

In realtà vi sono anche motivi tecnici e commerciali dietro a questi prodotti che per la maggior parte sono Spin off (giochi tratti da film o telefilm). Innanzitutto sono facili da realizzare, perché sfruttano routine già collaudate nei giochi precedenti... se togliamo la parte grafica, i giochi sono praticamente identici. In secondo luogo le software house spingono questo genere di prodotti perché possono contare su un sicuro successo di vendita dovuto alla notorietà del film/telemfilm e al desiderio dei ragazzi d'immedesimarsi nei loro eroi.

Ma per fortuna si sta già profilando un cambiamento di tendenza. Splendidi esempi di questo nuovo genere di prodotti interattivi, in cui il giocatore può impersonare un giornalista, un pilota, un esploratore, ci vengono per ora principalmente dall'America. La stessa Italvideo produrrà software in tal senso, rifiutando a priori il software "ultraviolento", anche se per motivi commerciali si è ancora costretti a utilizzare motivi "shock" per le immagini di copertina, dato che in questo mercato, nella maggior parte dei casi, il prodotto si vende più per quello che c'è in copertina che per il contenuto.

Un discorso a parte meritano certe follie di alcuni programmatori tedeschi e inglesi, che nel cosiddetto network-pirata stanno facendo circolare programmi demo per C-64 (di cui una copia è in mio possesso) i quali, per mezzo di house music digitalizzata e immagini pulsanti psichedeliche, pubblicizzano involontariamente (?) la nuova droga denominata Acid e incitano ai comportamenti tipici di chi assume questa droga. Programmi come questi li condanniamo fermamente e speriamo che nessun produttore europeo, per puri motivi di guadagno, voglia mai pubblicizzare giochi del genere.

Davide, pirata, anni 21: non ho mai visto i cosiddetti videogiochi nazisti e di conseguenza non posso esprimere un'opinione al riguardo, ma conosco bene i giochi violenti. Penso che questo genere di programmi abbia un'influenza molto marginale sui giocatori, comunque minore di quella che può avere un film o uno spettacolo televisivo. È certamente possibile che un ragazzino di otto o nove anni possa rimanere impressionato, e per questo sarebbe bene che i giochi venissero divisi per fasce d'età come accade per altri prodotti

Laura Maestri, responsabile della Lago snc: in qualità d'importatori e, soprattutto, di rivenditori tramite posta, siamo molto sensibili al problema e a un'eventuale crescita del fenomeno. I programmi ricreativi senza dubbio seguono una tendenza generale già sviluppatasi attraverso media molto più diffusi (cinema, televisione, musica...) volta a enfatizzare la spettacolarità delle situazioni, positive o negative che siano. Nell'ambito del gioco interattivo, l'immanicabile conseguenza è un'amplificazione degli effetti sonori e visivi che rendono il prodotto più appariscente e curioso. A nostro avviso, è comunque importante ricordare che, come per qualsiasi altro strumento di svago, la decisione determinante viene presa dal consumatore - o dai genitori, nel caso di utenti minorenni - che può scegliere tra programmi molto diversi, di cui è sempre disponibile un vasto assortimento.

Vitali, dettagliante: come negoziante non posso permettermi di censurare la mia merce, ma se i bambini cercano la violenza, nei videogiochi o in altri modi, è frutto di una determinata educazione ricevuta dalle famiglie. Non penso comunque che simulazioni violente possano spingere a una violenza reale, ma piuttosto che servano per sfogare l'aggressività che è in tutti noi. La recente polemica sui giochi nazisti o violenti è un effetto dell'eterna diffidenza verso le tecnologie non ancora assimilate. Non c'è nessuno che se la prende con i frigoriferi, ma solo perché ormai sono oggetti familiari a tutti e servono per riempirci la pancia.

John Holder, presidente della Leader Distribuzione srl: ci risulta che i cosiddetti nazi-game siano un fenomeno piuttosto marginale, che riguarda comunque il mercato tedesco. Sembra che siano adventure, e che quindi abbiano una forte componente testuale. Sono sicuramente poco interessanti per gli utenti italiani, a meno che non abbiano un'ottima conoscenza della lingua tedesca. I giochi del tipo *Jack the Ripper* sono pubblicati da software house piuttosto piccole, forse alla ricerca di qualcosa di scandaloso per farsi un po' di pubblicità (questo gioco non è stato importato dalla Leader). Tuttavia è difficile per noi sapere in anticipo il contenuto preciso di tutti i giochi lanciati sul mercato e dobbiamo dipendere dal buon senso delle software house.

Il mondo dei videogiochi non è un mondo a se stante. Se prendiamo in considerazione i venti giochi più venduti del 1988, constatiamo che ben 12 sono le conversioni per home computer di giochi coin op (quelli delle sale giochi), altri tre sono tratti da film (*Rambo III*, *Predator* e *Platoon*) e quattro sono giochi sportivi. È quindi ovvio che se i giochi da bar e i film hanno contenuti violenti, i videogame avranno la stessa caratteristica. Personalmente ritengo comunque che, per esempio, il gioco *Robo Cop* sia molto meno violento del film omonimo e sicuramente è molto meno scioccante perché il realismo di un computer è meno diretto che quello di un film.

In generale, credo che il gioco per computer sia un passatempo assolutamente innocuo, che ha l'unico "rischio" di essere a volte banale e poco costruttivo. A questo proposito però abbiamo notato un crescente interesse per i giochi più impegnativi come le simulazioni di volo, i giochi di strategia, quelli di ruolo e così via, tant'è vero che questo settore rappresenta il 25 per cento del nostro fatturato di software imparare a volare con un ottimo simulatore com'è *F-19 Stealth Fighter* o *F-16 Falcon*, oppure ricreare famose battaglie con *UMS* (Simulatore militare universale) può essere un esercizio mentale notevole oltre che una sfida stimolante e divertente ■

DAL NOSTRO INVIATO AD HANNOVER

CEBIT '89: TUTTE LE NOVITÀ DELLA COMMODORE

Un nuovo processore grafico per l'Amiga permette l'accesso alla libreria grafica TIGA, ma le novità più importanti sono le stazioni di lavoro professionali che hanno portato l'Amiga nelle università americane

di Michael Moosleitner

Il successo del CeBIT, tenutosi dall'8 al 15 marzo ad Hannover, nella Germania Occidentale, ha puntualmente riconfermato il primato della fiera tedesca nel settore informatico. Il numero dei visitatori accorsi da tutto il mondo ha per la prima volta superato la soglia del mezzo milione. Vista la crescente popolarità della manifestazione, la Commodore ha deciso di raddoppiare la propria superficie espositiva alla prossima edizione.

Lo stand Commodore

Per raggiungere lo stand della Commodore in mezzo alla marea di espositori e di visitatori era sufficiente seguire il flusso di giovani muniti di sacchetti con il marchio Amiga. Come un esercito d'infaticabili formiche, ragazze e ragazzi con il pallino dell'informatica infilava-

no nelle loro buste di plastica qualsiasi cosa capitava loro tra le mani. Erano piuttosto avviliti i rappresentanti degli espositori adiacenti alla Commodore, dove gruppetti di hacker si erano improvvisati campeggiatori, con tanto di zaini, lattine di birra e

cienza, la polizia ha annotato e fotografato, così come l'anno scorso aveva filmato alcuni noti ladri di software all'opera.

Le grandi novità esposte dalla Commodore non sembravano attirare i più giovani, salvo forse quelli dotati di molta fantasia. Si

trattava infatti di prodotti relativamente costosi e mirati a una fascia di utenza decisamente non casalinga.

Quello che abbiamo visto sul monitor collegato a un nuovo prototipo di scheda grafica per l'Amiga era un vero spettacolo di alta risoluzione. Il processore

grafico impiegato, il Texas TMS34010, consente di elaborare fino a sei milioni d'istruzioni al secondo e di utilizzare la libreria grafica standard TIGA. Grazie al chip Brooktree BT458 l'utente potrà lavorare con una palette di 16 milioni di colori. La risoluzio-



ne video sarà interamente programmabile, fino a un massimo di 1024 x 1024 pixel. Sarà possibile visualizzare contemporaneamente 256 colori diversi, più altri tre di overlay.

La tecnologia del transputer applicata all'Amiga ha iniziato a dare i suoi frutti. Due team di esperti di altrettanti centri di ricerca tedeschi hanno presentato alcune interessanti applicazioni basate sull'uso di stazioni di lavoro Amiga con transputer. *BRAGI* è un pacchetto grafico che consente di creare modelli di proteine dotate di nuove proprietà partendo dalle strutture tridimensionali di proteine note. Un altro sistema Amiga viene utilizzato per elaborare l'immensa mole di dati provenienti da un microscopio di tipo Atomic-Force o da uno scanner piezoelettrico. È così possibile effettuare alcuni tipi di ricerca sulle superfici dei cristalli in modo veloce ed economico. Anche il settore video ha iniziato ad adottare le nuove tecnologie: diversi programmatori hanno annunciato di lavorare a programmi di ray tracing e animazione basati sull'uso dei chip transputer.

L'Amiga nella rete

John Houston, presidente della Oxxi, è venuto personalmente dagli Stati Uniti per presentare l'ultima novità della sua ditta. Si tratta di una rete locale Novell basata su hardware Arcnet, completa di software e schede per Amiga 500 e 2000. Questa soluzione consente di collegare un grande numero di calcolatori e

periferiche tra loro. La configurazione presentata ad Hannover comprendeva un PC 40-III, che fungeva da file server, diversi PC, alcuni Amiga 2000 e un Amiga 500, nonché una stampante laser. Tutti gli Amiga erano in grado di effettuare il boot iniziale dalla rete, accedendo al disco rigido del PC 40-III come se fosse il loro, alla velocità di trasmissione di 2,5 Mbit per secondo. Abbiamo visto con i nostri occhi in che modo si può ottenere il velocissimo trasferimento su un PC, tramite *Deluxe Paint*, d'immagini create con i programmi grafici dell'Amiga. Numerose utility consentono lo scambio di messaggi tra gli operatori collegati alla rete, per esempio tramite quadri "pop up". Si prevede che la scheda Arcnet per l'Amiga 500, prodotta dalla Commodore, costerà meno di 500 dollari e permetterà di realizzare stazioni di lavoro "diskless" basate sull'Amiga 500, con ottime prestazioni e a un costo molto basso.

Da qualche mese alcune università americane stanno lavorando con il software *Amiga Unix V3.2*, che dovrebbe essere presto disponibile anche in Italia (si veda l'articolo "68020 e sistema operativo Unix per l'Amiga 2000" in questo stesso numero di *Commodore Gazette*).

Tra le altre novità minori per l'Amiga che abbiamo potuto vedere in esposizione ad Hannover, c'era un mouse ottico sviluppato da Dale Luck. Il nuovo mouse è totalmente compatibile dal punto di vista hardware con quello tradizionale. Ricordiamo che i vantaggi della tecnologia ottica sono principalmente la mancanza di

parti meccaniche, quindi di usura, e la maggiore precisione.

Ancora attesa per l'ECS

C'erano naturalmente anche diversi prodotti già descritti nei numeri 2/1988 e 1/1989 di *Commodore Gazette*, come l'unità a disco rigido più espansione RAM A590 per l'Amiga 500, che viene già prodotta in serie. In un caso, però, la Commodore ha preferito non esporre al pubblico i nuovi ritrovati: stiamo parlando delle macchine che dovrebbero avvalersi delle prestazioni dell'Enhanced Chip Set... sembra infatti che i nuovi chip per l'Amiga non siano completamente privi di bug. Ricordiamo che l'ECS dovrebbe consentire di superare le attuali limitazioni di memoria chip e risoluzione video, e dovrebbe migliorare la velocità di alcune funzioni grafiche.

Per molti giovani americani il CeBIT, come sempre, è stato qualcosa di più di una semplice fiera. Qualcuno ha voluto provare l'ebbrezza di noleggiare una Mercedes e lanciarla sulle autostrade prive di limiti di velocità, cercando di riprodurre le evoluzioni dei veicoli antigravità di *Guerre stellari*. Inevitabilmente qualche astronave ha finito la sua corsa in un fosso, e c'è stato chi ha scambiato i binari della metropolitana leggera per chissà quale rampa di lancio. Ma sono situazioni piuttosto consuete in questi incontri, e la colpa, se di colpa si può parlare, andrebbe addebitata alla gustosa birra locale.

LA IHT GRUPPO EDITORIALE CERCA

REDATTORI TECNICI PER LA PROPRIA REDAZIONE

La IHT CERCA UN REDATTORE DA INSERIRE NEI QUADRI DEL PERSONALE La località di lavoro è Milano. INVIARE CURRICULUM DETAGLIATO

AGENTI PUBBLICITARI

PER LA TESTATA COMMODORE GAZETTE LA IHT CERCA AGENTI PUBBLICITARI O AGENZIE DI VENDITA DEGLI SPAZI PUBBLICITARI.

TRADUTTORI

LA DIVISIONE LIBRI CERCA TRADUTTORI AI QUALI AFFIDARE LA TRADUZIONE DALL'INGLESE ALL'ITALIANO DI TESTI TECNICI SULL'AMIGA INVIARE CURRICULUM DETAGLIATO

Gli interessati possono scrivere a:

IHT GRUPPO EDITORIALE - Via MONTE NAPOLEONE, 9 - 20121 MILANO

UN AMBIENTE DI PROGRAMMAZIONE EFFICIENTE

*Come predisporre l'Amiga e l'ambiente di programmazione
a seconda della configurazione hardware di cui si dispone.
Un file comandi studiato per la compilazione automatica*

di Eugène P. Mortimore

Nella scorsa puntata abbiamo preso in esame il pacchetto di programmazione Manx, analizzandone tutti i file di input e di output che intervengono nelle varie fasi della programmazione. L'analisi che abbiamo svolto ci ha offerto un panorama completo di questo pacchetto, e ci ha mostrato a quale livello di complessità può giungere un sistema per la programmazione in C. A questo punto occorre integrare il pacchetto Manx con l'Amiga, e questo è un compito molto delicato se si desidera lavorare in un ambiente di programmazione efficiente.

In questo articolo vengono fornite alcune direttive sulla procedura d'integrazione, facendo l'ipotesi che l'utente disponga di un Amiga (di qualsiasi tipo) con due disk drive. Una suddivisione ottimale dei file sui dischi richiede infatti almeno una configurazione dotata di due disk drive, per disporre di tutto lo spazio necessario ed evitare inutili attese (meglio ancora sarebbe avere un disco virtuale abbastanza grande, o addirittura un hard disk). Questa soluzione, comunque, può essere adattata anche per un solo disk drive, sacrificando alcuni tool che rendono più semplice e più veloce il lavoro di programmazione.

Supporremo inoltre che si vogliano avere a disposizione tutti i file del pacchetto Manx e che debba restare lo spazio necessario per i file sorgente, e ancora un po' di spazio per tutte le utility di uso più frequente; parleremo poi di un metodo per rendere più efficiente l'uso dei disk drive e di come si può alterare la configurazione del sistema per adattarla ad altre esigenze. Infine, analizzeremo un comodo file comandi che permette di trasformare automaticamente i file sorgente in file eseguibili, verificando nel frattempo anche se è presente qualche errore.

Il disk drive di serie DF0:

Nel disk drive DF0: dev'essere presente un disco contenente le directory c, l, devs, s, t, fonts, e libs. Perché il sistema le sappia individuare occorre che siano assegnate a particolari directory logiche. In particolare, occorre che la c sia assegnata a C:, la l sia assegnata a L:, la devs sia assegnata a DEVS:, e così via. Per default il sistema compie queste assegnazioni durante il boot (se si digita il comando ASSIGN senza indicare argomenti si ottiene la lista delle assegnazioni), ma se per qualche ragione si desidera cambiare la posizione di una di queste directory o anche solo il suo nome, occorre ripeterne l'assegnazione tramite il comando ASSIGN. Nel corso dell'articolo si userà il sostantivo directory sia per individuare le directory fisiche (c, devs...) sia per individuare le directory logiche (C:, DEVS:...); non ci sarà comunque ambiguità dal momento che le seconde hanno sempre nomi completamente maiuscoli e sono seguite dal carattere ":". Vediamo quali sono i file che ognuna di queste directory dovrebbe contenere.

- La directory C: deve contenere i comandi dell'AmigaDOS, come per esempio COPY, DIR e tutti gli altri tool presenti nella directory c del disco sistema. Disponendo dei pacchetti WShell o AREXX, c'è la possibilità di sostituire i comandi standard con quelli previsti dal progetto ARP (AmigaDOS Replacement Project, per altre informazioni vedere *Commodore Gazette* n. 2/88, pag. 114), i quali, oltre ad aggiungere nuove opzioni, consentono di risparmiare spazio e tempo di esecuzione.

- La directory L: deve contenere tutti gli handler di sistema, come il Disk-Validator che si occupa di effettuare il validate di ogni disco

inserito in un disk drive, oppure come il Port-Handler che gestisce le operazioni attraverso le porte seriale e parallela dell'Amiga (non si commetta l'errore di confondere gli handler con i dispositivi di I/O come TrackDisk e Serial). Anche eventuali handler non appartenenti al software sistema originale devono essere inseriti in questa directory: ne è un esempio ConMan, presente nel pacchetto WShell, che serve a migliorare le prestazioni del dispositivo Console, oppure Newcon-Handler presente nella versione 1.3 del Workbench.

- La directory DEVS: deve contenere le subdirectory keymaps, printers e clipboards, nonché le librerie shared per la gestione dei dispositivi, come serial.device e narrator.device. Queste librerie costituiscono i dispositivi di I/O non residenti, che il sistema carica in memoria quando all'interno di un task viene eseguita la funzione OpenDevice con il nome del particolare dispositivo come argomento. Il sistema si aspetta sempre che siano contenute nella directory DEVS:, quando le cerca sul disco. In questa stessa directory devono risiedere anche i file system-configuration e MountList. MountList riveste una particolare importanza nell'attivazione dei dispositivi logici e degli handler, dal momento che contiene tutti i loro parametri di attivazione.

- La directory S: è importante perché il sistema si aspetta di trovarvi la startup-sequence, cioè il file comandi che viene eseguito durante il boot. In questa directory devono anche risiedere tutti i file comandi che si vogliono impiegare tramite il comando EXECUTE, e altri file comandi previsti da particolari applicazioni. Per esempio il CLI, quando viene attivato, prima di fare apparire il prompt verifica se nella directory S: esiste un file comandi di nome CLI-Startup, nel qual caso lo esegue. La stessa operazione viene compiuta dallo Shell, che manda in esecuzione il file Shell-Startup se è presente. In questo articolo punteremo la nostra attenzione sul file comandi startup-sequence per il boot del sistema, e sul file ccln che rende automatica la compilazione e la fase di link. Startup-WShell, che è il file di startup dell'interfaccia linea comando WShell, viene invece trattato in un altro articolo che troverete in questo stesso numero della rivista.

- La directory T: conterrà tutti i file temporanei prodotti dal sistema e dalle applicazioni, come per esempio il file di sicurezza che un editor opportunamente configurato potrebbe salvare periodicamente. Anche se la sua presenza è indispensabile, non è necessario sapere altro sul suo contenuto.

- La directory FONTS: deve contenere tutte le fonti-carattere di cui si farà uso. Per esempio, se si usa l'editor TxEPlus si può collocare in questa directory la fonte Siesta distribuita con il programma. Sebbene le fonti-carattere non portino via molto spazio, è ugualmente opportuno lasciare la directory FONTS: vuota se non occorrono fonti diverse da quella di default residente su ROM. Si noti che abbiamo citato come esempio l'editor di

testi TxEPlus, anziché CygnusEd Professional, in quanto le sue dimensioni sono inferiori e pur offrendo meno opzioni è abbastanza sofisticato da permettere di redigere un programma in C senza difficoltà. Se si possiedono tre disk drive o un hard disk, comunque, è preferibile CygnusEd Professional. L'editor TxEPlus verrà analizzato in uno dei prossimi articoli.

- La directory LIBS:, il cui nome deriva da "LIBraries for Sharing among tasks or processes", riveste un ruolo fondamentale nella programmazione e nell'uso delle applicazioni. Al suo interno, infatti, il sistema si aspetta di trovare le librerie non residenti su ROM alle quali i task accedono eseguendo la funzione OpenLibrary. Per distinguere dalle librerie di tipo linked, che intervengono nelle fasi di link, vengono chiamate librerie shared (si ricordi che le librerie shared sono collezioni di funzioni che possono essere condivise da più task). Oltre alle librerie previste dall'Amiga, come translator.library e icon.library, è possibile impiegarne anche altre, magari previste da particolari applicazioni. Comunque, indipendentemente dalla loro origine, le librerie shared (a parte quelle relative ai dispositivi) devono sempre risiedere nella directory LIBS: perché il sistema possa individuarle.

Per esempio, il pacchetto WShell utilizza due librerie shared non standard, WShell.library e consolehandler.library, che devono dunque essere copiate dal disco originale di WShell nella directory LIBS:. Anche il sistema AREXX utilizza due librerie shared non standard, rexxsupport.library e rexxsyslib.library, che devono essere copiate in DEVS: per consentire il funzionamento delle macro e dei programmi gestiti da AREXX.

- La directory include. Il compilatore della Manx anziché impiegare una directory logica utilizza la variabile globale INCLUDE per individuare nel sistema la directory che contiene i file INCLUDE (le variabili globali vengono impostate con il comando SET previsto dal pacchetto del compilatore). In questa directory devono risiedere tutti i file INCLUDE che intervengono nella compilazione dei sorgenti in C. Normalmente questi file sono ulteriormente ripartiti in subdirectory come exec o devices. Non è detto che tutti i file INCLUDE presenti nel disco SYS1 del compilatore debbano essere copiati nella directory include, mentre invece può presentarsi la necessità di aggiungerne di nuovi realizzati appositamente per i propri sorgenti. Per esempio, in quasi tutti i sorgenti di programmi che impiegano qualche risorsa dell'Amiga appare la direttiva #include <exec/types>, mentre sono molto meno quelli nella quale appare la direttiva #include <devices/audio.h>.

L'insieme dei file INCLUDE commentati previsti dalla Commodore occupa molto spazio su disco, ed è quindi opportuno limitare la loro presenza a quelli indispensabili (ma ogni volta che si affronta un nuovo sorgente la scelta va rifatta). Se invece si può fare a meno dei commenti, come accade di

solito, lo spazio occupato dall'intero insieme dei file INCLUDE previsto dalla Commodore diventa di proporzioni ragionevoli, e non occorre sacrificare nessun file.

– La directory radice deve contenere prima di tutto le directory elencate (che diventano quindi subdirectory), quindi i file sorgente in C, i moduli oggetto prodotti dall'assembler, i programmi eseguibili e tutti gli altri file utili per il debug derivanti dal processo di compilazione, come i file .sym e .dbg del sistema Manx esaminati nel precedente articolo della serie.

Concluse le operazioni preliminari per la preparazione del disco contenuto nel disk drive di serie, dovrebbe rimanere nella directory radice ancora abbastanza spazio (circa il 25 per cento degli 880K disponibili) per contenere i file necessari allo sviluppo di un programma di medie dimensioni. Una migliore organizzazione dello spazio su disco può essere ottenuta soltanto analizzando le dimensioni e le caratteristiche del programma e dei relativi file di supporto.

Il disk drive aggiuntivo

Nel disk drive aggiuntivo dev'essere presente un secondo disco nel quale collocheremo gli altri file necessari alla programmazione. Prima di tutto i tool del pacchetto di programmazione, come il compilatore *cc*, l'assembler *as*, il linker *ln* e il *Source Level Debugger* (che su disco appare con il nome *sdb*), poi tutte le librerie linked previste dal pacchetto della Manx. Ricordiamo che queste librerie si trovano distribuite nelle directory *lib* dei tre dischi *SYS1*, *SYS2* e *SYS3* che compongono il pacchetto di programmazione *Aztec C68K* della Manx. L'esatto numero di librerie linked richieste varia a seconda del programma da costruire e tenendo conto che alcune librerie occupano molto spazio è consigliabile un'attenta selezione.

Si tenga tuttavia presente che se il disco sistema (quello che risiede in *DF0:*) è stato opportunamente privato di tutti i file che non sono strettamente necessari, magari riducendo la presenza di file INCLUDE allo stretto indispensabile, si può anche avere lo spazio sufficiente per copiare i tool di programmazione (come il compilatore e il linker) nella directory *C:*, lasciando più spazio sul disco che andrà nel disk drive aggiuntivo.

La configurazione che stiamo illustrando ci permetterà di lavorare anche in Assembly, all'interno dei sorgenti in C oppure separatamente, in quanto sul secondo disco memorizziamo anche i file INCLUDE in linguaggio Assembly, quelli dotati dell'estensione ".i" che nel pacchetto della Manx sono raccolti all'interno della directory *asm* del disco *SYS2*.

Infine, nel secondo disco possiamo memorizzare anche quei tool di sistema utili durante la fase di sviluppo dei programmi, come *lb*, se capita con una certa frequenza di creare librerie linked personalizzate (si noti comunque che le librerie

linked possono anche essere considerate come normali moduli oggetto).

Scendere troppo nei dettagli potrebbe confondere le idee, quindi vi consigliamo di studiare il precedente articolo di questa serie e la documentazione del compilatore in vostro possesso per capire quali altri file possono servire a seconda dei programmi da sviluppare. In ogni caso, anche dopo aver copiato altri file, nel secondo disco resterà sicuramente parecchio spazio, che potrà essere utilizzato per copiarvi i file dei programmi o dei moduli che sono giunti alla versione definitiva e non hanno bisogno di ulteriori modifiche. Questa operazione di pulizia del disco sistema, il disco su cui si svolgono quasi tutte le nostre operazioni di programmazione, si dovrebbe effettuare abbastanza spesso.

Organizzando il disco sistema e il secondo disco nel modo illustrato, si riesce a sfruttare adeguatamente la configurazione con due disk drive senza dover sacrificare nessuno strumento di programmazione.

I file di startup per la programmazione

Il Listato 1 riporta una tipica versione di startup-sequence utilizzabile in un sistema di programmazione in C. Questo file suggerisce in che modo si può operare perché l'Amiga, concluso il boot, si presenti al programmatore nelle condizioni di maggior efficienza.

In questo contesto facciamo ancora riferimento a un sistema con due disk drive, nel quale siano disponibili i pacchetti *AREXX* e *WShell*. Per chi non dispone delle citate espansioni, spiegheremo come adattare il Listato 1 perché sia eseguibile dallo Shell del *Workbench 1.3*.

Nel file sono stati ben evidenziati i gruppi di comandi non standard che si riferiscono a un particolare pacchetto e che quindi devono essere eliminati da chi non ne dispone. Altri comandi si possono invece adattare per funzionare all'interno dello Shell, come *ALIAS* e *RESI* previsti dall'interfaccia utente *WShell*. Per quanto riguarda il comando *RESI*, è sufficiente cambiare il suo nome in *RESIDENT*, mentre con il comando *ALIAS* occorre una procedura più complicata. Oltre ad avere una sintassi diversa rispetto a quella prevista da *WShell*, infatti, l'*ALIAS* disponibile con lo Shell del *Workbench 1.3* non è neanche un comando residente su disco, cioè non si trova in nessuna delle directory del disco sistema. È contenuto invece nello Shell, e come tale è eseguibile soltanto dallo Shell.

Veniamo alle operazioni che devono svolgere in pratica tutti coloro che dispongono soltanto dello Shell di sistema. 1) Inserire nella propria startup-sequence tutti i comandi elencati nel Listato 1 che non richiedono altri particolari pacchetti (eccetto *ALIAS*); eseguendo questa operazione occorre ricordarsi di cambiare i comandi *RESI* in *RESIDENT*. 2) Inserire al termine della propria startup-

sequence, prima del comando ENDCLI > NIL.; il comando NEWSHELL, di modo che durante l'esecuzione sequenziale dei comandi venga aperto uno Shell prima della chiusura del CLI aperto dal sistema. 3) Memorizzare infine nella directory s del proprio disco sistema un file di nome Shell-Startup contenente tutti i comandi ALIAS del listato che si ritengono necessari, ricordando che i simboli di uguale (=) previsti dalla sintassi WShell devono essere trasformati negli spazi previsti dalla sintassi Shell (Shell-Startup è il file che lo Shell esegue per default quando viene mandato in esecuzione).

Con questa procedura si può utilizzare la startup-sequence del Listato 1 anche se non si dispone dell'interfaccia utente WShell che, come si è visto, differisce molto poco dallo Shell introdotto dal *Workbench 1.3* (lo Shell oltre a possedere tutte le caratteristiche di un CLI, aggiunge il piping, l'editor di linea con un buffer storico dei comandi impartiti, la possibilità di trasformare alcuni comandi in comandi residenti, e infine il comando ALIAS).

Passiamo ora a descrivere la startup-sequence del Listato 1. Dopo un'istruzione ECHO il cui unico scopo è quello di dare il benvenuto all'utente, viene fissato a 100 il massimo codice d'errore che può essere tollerato dal sistema durante l'esecuzione dei comandi: se uno dei comandi presenti nella startup-sequence restituisce un codice d'errore maggiore o uguale a 100, l'esecuzione del file comandi viene interrotta. Questo numero è arbitrario, e si può scegliere il valore che si preferisce.

L'operazione successiva stabilisce in 6 mila byte la dimensione di default dello stack del CLI. Si tratta della dimensione che verrà impiegata dall'interfaccia linea comando attiva, e da tutti i processi che da essa verranno mandati in esecuzione. Si noti che il comando STACK agisce prescindendo dal tipo d'interfaccia linea comando (CLI, Shell, WShell...). Seimila byte è un buon compromesso per essere sicuri che non si riempirà mai del tutto lo stack sia durante la procedura di startup sia durante le successive fasi di compilazione e di link. Uno stack di queste dimensioni dovrebbe mettere a disposizione per i tool del compilatore abbastanza spazio per la memorizzazione delle loro variabili temporanee; se questa quantità si dovesse rivelare insufficiente, è bene ricorrere al comando STACK per aumentarla (si noti che se si verifica l'overflow dello stack durante la compilazione o nella fase di link, il sistema si blocca).

Con le tre istruzioni successive si assegna un nome più corto ai dispositivi logici dei tre disk drive principali: "a:" al posto di "DF0:", "b:" al posto di "DF1:" e "r:" al posto di "RAM:". In questo modo ci si può riferire ai disk drive più facilmente, e comunque la vecchia denominazione resta sempre valida. Più avanti nel file, facendo uso degli alias, le cose diventano ancora più semplici. Un'altra operazione che viene compiuta in questa fase è la creazione di tre subdirectory (c, s, t) nel

RAM disk, nelle quali verranno memorizzati alcuni comandi e tool al fine di rendere più veloce la loro esecuzione.

Cerchiamo ora di specificare il percorso di ricerca (path) attraverso il quale l'AmigaDOS deve cercare i file da eseguire: stabiliamo come prima possibilità la directory c del RAM disk, la più veloce da esplorare, quindi la directory radice RAM:, quindi la directory c del disco sistema... e così via sino alla directory radice del disco aggiuntivo. Di solito, infatti, tutti i file eseguibili, compresi i tool del pacchetto di programmazione, vengono sistemati nella directory c del disco sistema (o del disco virtuale se si dispone di sufficiente memoria), oppure nella directory c del secondo disco. Ovviamente, la posizione che assicura la massima velocità di ricerca e di esecuzione per tutti i tool è il RAM disk, ma questa scelta riduce la memoria disponibile nel sistema.

La copia nel RAM disk dei file eseguibili

Le cinque istruzioni successive copiano altrettanti file nella directory c del RAM disk (i file che vengono usati più spesso). Il primo è l'editor, tramite il quale si redigono i file sorgente: supponendo che s'intenda utilizzare *TxEdition Plus*, il file comandi lo copia dalla directory c del disk drive a: e gli cambia nome, definendolo "ee". Si potranno così modificare rapidamente i file sorgente in qualsiasi momento se ne presenti la necessità. Naturalmente è possibile usare un altro editor al posto di *TxEdition Plus*, come per esempio *Z della Manx*, *LSE della Lattice* o il già citato *CygnusEd Professional*, e in ogni caso ne verrà mutato il nome in "ee".

La medesima operazione viene compiuta sui comandi COPY, DELETE e LIST. Infine viene copiato il tool grep del pacchetto Manx, che si utilizza quando si devono cercare stringhe di caratteri all'interno dei file sorgente.

È importante notare che attraverso l'uso del WShell o dello Shell, i comandi COPY, DELETE e LIST, possono eventualmente essere trasformati in residenti per evitare di copiarli nel RAM disk. A questo proposito ricordiamo che esistono tre condizioni fondamentali perché un comando possa diventare residente: occorre che non si automodifichi durante l'esecuzione, che reinizializzi tutte le sue variabili ogni volta che viene eseguito, e che possa agire in maniera multitasking. Se si possiedono ancora i comandi della versione 1.2, non ci sono indicazioni ufficiali su quali possono diventare residenti e quali no, e quindi occorre sperimentare ogni singolo caso. Se invece si possiede la versione 1.3 del *Workbench*, è sufficiente impartire il comando LIST C: e scartare tutti i comandi che non hanno il flag p impostato.

La definizione delle variabili di sistema

Ora daremo un valore ad alcune delle variabili di sistema utilizzate durante il processo di compila-

zione. Si fa riferimento ancora una volta al sistema Manx, ma con il Lattice non vi sono differenze se non per il nome delle variabili.

Una delle caratteristiche più interessanti del sistema operativo dell'Amiga è la possibilità di specificare le directory logiche all'interno delle quali dev'essere cercato ciascun file. Per caricare una libreria shared, per esempio, il sistema accede alla directory logica LIBS:, dove si aspetta di trovarla. L'utente ha comunque la facoltà di cambiare la directory fisica corrispondente a LIBS:, in modo che il sistema acceda a una directory piuttosto che a un'altra. Questo vale per tutti i file che il sistema operativo deve individuare.

Il pacchetto Manx, tuttavia, per ottimizzare la ricerca dei file che intervengono nelle varie fasi della creazione di un programma non impiega i dispositivi logici, ma le variabili globali che l'utente o la startup-sequence hanno impostato tramite il comando non standard SET. Si tratta di variabili che non interferiscono con i dispositivi logici considerati dal sistema operativo, e che quindi influiscono solo sui tool del pacchetto Manx.

Nel nostro file comandi di startup, i primi due comandi SET indicano all'editor Z che i file z.opt, il file delle opzioni, e global.tags, il file dei tags globali, si trovano nel RAM disk. Dal momento che pochi programmatori si servono di questo antiquato editor di derivazione Unix, questi due comandi SET si possono omettere quasi sempre.

Se si fa uso del *Gimple Lint syntax checker*, la successiva istruzione serve per assegnare alla variabile FUNCLIST il percorso di ricerca relativo a un utile file che potremmo chiamare file-sintassi. Nel pacchetto del tool *Gimple Lint* è compreso un esempio di questo file che contiene tutte le funzioni normalmente usate durante la stesura di programmi in C con il pacchetto Manx, per ognuna delle quali riporta la sintassi di chiamata (cioè il numero di argomenti e i loro tipi). Il programmatore può comunque creare un proprio file-sintassi per le funzioni che definisce all'interno di un suo programma. Nel nostro caso questo file viene collocato nel RAM disk con il nome manx.c: in questo modo il syntax checker ha la possibilità di controllare la correttezza di ogni chiamata di funzione all'interno del programma.

Con l'istruzione successiva si comunica al sistema che i file INCLUDE si trovano nella directory include del disco sistema. Questo per quanto riguarda i file INCLUDE ".h" in linguaggio C; se però desideriamo lavorare anche in Assembly, può essere opportuno specificare attraverso il comando SET anche la posizione dei file INCLUDE ".i".

La successiva linea del file comandi stabilisce in quale disco il compilatore scriverà i suoi file temporanei in Assembly. Nel nostro caso si tratta del RAM disk, nella subdirectory t. Questa scelta contribuisce ad accelerare il processo di compilazione e non spreca memoria preziosa, in quanto il file viene cancellato subito dopo essere stato letto dall'assembler: basta soltanto che ci sia abbastanza

memoria per soddisfare la sua momentanea richiesta.

Il comando successivo indica al linker dove vanno cercate le librerie linked. Nel nostro caso la ricerca inizia sul disco sistema, nella directory lib, poi nella stessa directory del secondo disco nell'evenienza che non tutti i file di libreria siano stati spostati nel disco sistema, (attenzione: le librerie linked non sono le librerie shared, che si trovano sempre nella directory LIBS:).

Con l'ultimo comando SET viene stabilito che il file d'inizializzazione del debugger db, cioè il file .dbinit, si trova nella directory DF0:s. Si ricordi che db è il debugger che viene distribuito con il pacchetto della Manx, molto meno sofisticato del noto *Source Level Debugger*. Se si dispone di quest'ultimo, il comando può essere ommesso.

A questo punto troviamo due comandi che rendono attivi i miglioramenti di cui è portatore il *TxED Plus*: questo è possibile se si sono copiati i file "BlitzDisk" e "FastFonts" dal disco dell'editor in una delle directory nelle quali il sistema cerca i file eseguibili. Il programma "FastFonts" accelera la visualizzazione dei testi, per esempio durante le fasi di scroll, in ambiente *TxED Plus*; non è indispensabile, ma migliora notevolmente le operazioni di correzione.

Invece il programma "BlitzDisk" crea un buffer di cache per i disk drive, grazie al quale tutte le operazioni che coinvolgono i dischi vengono sveltite e si eliminano le consuete attese dopo aver impartito comandi come DIR o INFO. La prima volta che si richiede la visualizzazione della directory, infatti, le informazioni a essa relative vengono memorizzate nell'area di cache, così le successive richieste verranno eseguite molto più velocemente in quanto la lettura avverrà dalla RAM.

L'interfaccia utente WShell

Con i tre comandi successivi chiamiamo in causa il pacchetto *WShell* e il nuovo handler di console con ConMan in esso contenuto. Il primo comando, SetExecute, fa in modo che il comando EXECUTE dell'AmigaDOS funzioni correttamente con i tool del sistema *WShell*. In questo modo, una volta aperta una nuova interfaccia utente con il comando NEWWSH, tutti i file comandi dell'AmigaDOS funzioneranno correttamente.

Il comando successivo attiva l'handler di console ConMan. Questo handler consente di sostituire il consueto dispositivo logico CON: con il nuovo CNC:, che aumenta le potenzialità delle finestre CLI, permettendoci di richiamare comandi già impartiti, di utilizzare gli alias e il piping (l'output di un programma inviato direttamente in input a un altro programma) e così via. Dopo l'esecuzione di questo comando, tutti gli altri shell che verranno aperti con NEWWSH faranno automaticamente uso delle potenzialità dell'handler ConMan. Di fatto questo handler è molto simile al

nuovo handler Newcon-Handler disponibile con la versione 1.3 del *Workbench* (è a questo particolare tipo di handler che dobbiamo la command history, un buffer che ci permette d'impartire comandi già eseguiti senza doverli riscrivere).

La successiva istruzione attiva una nuova interfaccia linea comando *WShell*. Si noti che fino a questo punto è stato aperto solo un normale CLI, quello inizializzato dall'AmigaDOS durante il boot della macchina. Questo particolare assume maggiore importanza se si dispone soltanto dello Shell in dotazione al *Workbench 1.3*, in quanto comandi come ALIAS non possono essere eseguiti se non dall'interno dello Shell.

Con le tre istruzioni seguenti, i file contenenti le variabili di sistema dalla directory `s:env` del disco sistema vengono copiati nella directory `RAM:env`, e si assegna a quest'ultima il dispositivo logico `ENV:`. Il sistema *WShell*, al pari dello Shell, è progettato per ricercare automaticamente nella directory `ENV:` i file contenenti le variabili globali relative a ciascuna finestra. Se si possiede il *WShell*, la directory `RAM:env` contiene per esempio un file di nome `TitleBar` che controlla quali informazioni devono apparire nella barra titolo della finestra. Un possibile contenuto di questo file è il seguente: `C-Dev TN=%n TI=%i EI=%e RC=%r EC=%y SS=%s CD=>>>%c`. Queste specifiche assegnano alla finestra il nome "C-Dev" e inoltre visualizzano i seguenti dati: 1) il numero del task dell'Exec in esecuzione nel *WShell* aperto, preceduto dalla dicitura "TN="; 2) l'identificatore di quel task, TI; 3) il tempo trascorso dall'ultima operazione compiuta dallo shell, ET, utile soprattutto nei programmi in cui il tempo d'esecuzione è un fattore significativo; 4) il codice d'errore dell'ultimo comando eseguito, RC, e di quello attuale, EC; 5) l'ampiezza dello stack, SS, normalmente 6 mila byte; 6) l'attuale directory, preceduta dai caratteri "CD=". Così l'intestazione della finestra riflette continuamente lo stato del sistema di programmazione, a mano a mano che si prosegue nell'elaborazione dei comandi. Si noti in particolare che la directory appare sempre come ultima voce nella barra titolo, allo scopo di mantenerci costantemente informati su questo parametro che cambia spessissimo durante l'attività del computer.

La creazione dei sinonimi

Segue poi una lunga serie di comandi, che ricorrono alle capacità di definizione degli alias del sistema *WShell*, al fine di personalizzare ulteriormente il sistema e facilitarne ancora l'impiego. L'elenco è piuttosto lungo e segue volutamente una linea di genericità, per tentare di soddisfare le esigenze più varie. Alcuni alias rappresentano strade diverse per raggiungere gli stessi scopi, e quindi bisognerà scegliere quelli che si preferiscono. Questo vale per tutte le possibilità previste: se alcune non sembrano interessanti, possono essere

tranquillamente cancellate. Si ricordi che i comandi ALIAS elencati possono essere impiegati anche se si dispone solo dello Shell, a patto però che vengano inseriti nel file comandi Shell-Startup e che tutti i caratteri "=" vengano sostituiti con uno spazio.

Ora discuteremo alcuni dei nomi da noi assegnati agli alias. I primi tre sono abbreviazioni per selezionare rapidamente il disk drive; se si desidera per esempio che venga eseguito il comando `CD DF1:`, è sufficiente premere il carattere "0" e il RETURN, un metodo senz'altro più rapido che digitare sette caratteri. Agli altri due disk possiamo accedere con la stessa facilità, digitando "1" per il secondo disk drive e "2" per il RAM disk.

Continuando nell'esame dell'elenco di nomi, scopriamo altri utili alias: "ess" per esaminare e/o modificare la startup-sequence; "daf" e "dbf" per memorizzare la directory completa del disco contenuto in `DF0:` (`DF1:`) in un file chiamato `RAM:dira` (`RAM:dirb`); quindi i comandi "dap" e "dbp", che servono a stampare la directory di un disco. Come si può vedere, la maggior parte degli altri alias sono abbreviazioni dei nomi dei file eseguibili presenti nella directory `c` del disco sistema.

La successiva serie d'istruzioni all'interno del file di startup è un gruppo di comandi RESI. Perché possa essere eseguito, il comando RESI dev'essere copiato dal disco sistema del pacchetto *WShell* nella directory `c` del disco sistema che abbiamo opportunamente preparato (si ricordi che se si dispone soltanto dello Shell tale comando dev'essere indicato con il nome RESIDENT). Lo scopo del comando RESI è quello di rendere residente in memoria un comando, abbreviando così il tempo di risposta del sistema quando il comando viene impartito.

Nel nostro esempio abbiamo trasformato in residenti i comandi più comuni, come DIR, CD, MAKEDIR, PATH, ALIAS e alcuni altri. Si noti che se si dispone solo dello Shell, il comando ALIAS è sempre residente, e quindi non si può ridefinirlo come tale, anche perché non appare in nessuna directory del disco sistema; inoltre, si tenga presente che i comandi CONMAN, CDTITLE, HISTORY, NEWWSH, e RUNWSH sono specifici del pacchetto *WShell*, e non sono presenti nel *Workbench 1.3*. Come per gli alias, l'elenco di comandi RESI è soltanto un suggerimento, e ognuno può costruire una propria gamma di comandi.

Il resto di questo file comandi è costituito da quattro istruzioni. La prima ordina all'handler ConMan di leggere e caricare in memoria un determinato file comandi storico, denominato ProgrammingHistory, il quale contiene la sequenza dei comandi impartiti da tastiera in una fase precedente. Per esempio, si supponga di essere impegnati nella creazione di un programma in C, composto da tre file sorgenti denominati `f0.c`, `f1.c` e `f2.c`; supponiamo inoltre che nella precedente

sessione di lavoro questi file siano stati sottoposti a edit usando comandi tipo "ee f0.c", "ee f1.c" e così via. Se era attivo l'handler ConMan, il buffer storico ha immagazzinato i comandi a mano a mano che venivano impartiti. Se alla fine del lavoro abbiamo ricordato di salvare questo buffer nel file ProgrammingHistory, ora è sufficiente rileggere il file per ricreare nel nuovo buffer storico la stessa sequenza di comandi. A questo punto basta utilizzare i tasti che muovono il cursore verso l'alto e verso il basso per richiamare tutti i comandi impartiti nella sessione precedente, e questo rappresenta un notevole risparmio di tempo se i comandi necessari in ogni sessione di lavoro sono più o meno gli stessi. Si noti che con il solo Shell non è possibile caricare o salvare su disco il contenuto del buffer storico.

Le due istruzioni successive sono riservate a chi dispone del pacchetto AREXX. La prima assegna la directory df0:rexx al dispositivo logico REXX. In questa directory devono risiedere tutti i programmi batch relativi al pacchetto AREXX. La seconda istruzione, REXXMAST, attiva l'interprete residente dei programmi batch in standard AREXX: quando tale interprete è attivo, si può usare il comando RX per mandare in esecuzione i programmi batch.

L'ultimo comando del file di startup è ovviamente ENDCLI > NIL:, che chiude il CLI aperto dall'AmigaDOS durante il boot della macchina. Dopo l'esecuzione di questo comando, l'unica interfaccia linea comando che rimane è quella aperta dal comando NEWWSH (o NEWSHELL se si dispone solo dello Shell) all'inizio del file di startup.

Il file comandi per la compilazione e la fase di link

Il Listato 2 propone un utile file comandi che rende automatica la trasformazione di un file sorgente in file eseguibile, preoccupandosi di visualizzare, tramite un editor, gli errori che eventualmente si verificano durante la compilazione o la fase di link. Questo file comandi può essere usato soltanto se il programma da compilare è contenuto in un unico file sorgente. Non si tratta comunque di una grossa limitazione, dal momento che i primi programmi che si scrivono rientrano quasi sempre nelle dimensioni di un unico file sorgente.

Questo file comandi è stato scritto esplicitamente per il pacchetto Manx. Per eseguirlo, supponendo che si chiami ccln e che il file sorgente si chiami Programma.c, occorre impartire la seguente linea comando:

```
EXECUTE ccln Programma
```

La prima operazione che viene compiuta è memorizzare nella variabile file l'argomento che l'utente ha indicato dopo il nome del file comandi. Se l'utente non ha indicato alcun nome come

argomento, viene visualizzata una stringa e il controllo viene ceduto all'etichetta END, corrispondente alla fine del file comandi. Segue un altro controllo che verifica se il file indicato come argomento, seguito dall'estensione ".c", è presente su disco. Se non lo è l'utente viene avvisato e il sistema considera conclusa l'esecuzione del file comandi.

La successiva operazione è la cancellazione del file AztecC.err, se è presente. Questo file viene generato dal compilatore quando si indica l'opzione -Q e si verificano uno o più errori durante la compilazione. Il motivo per cui viene cancellato è che il nostro file comandi, come vedremo fra poco, scopre se la compilazione ha avuto successo proprio andando a verificare se il file esiste oppure no.

Compiuta questa serie di controlli, il flusso prosegue con l'esecuzione di un comando ECHO per visualizzare un prompt e con la compilazione del file sorgente. Anzitutto, l'output del compilatore viene ridiretto verso il file RAM:cc.w, esaminando il quale l'utente può controllare se durante la compilazione sono stati generati i cosiddetti warning (avvertimenti), cioè condizioni che non sono veri e propri errori ai fini della compilazione, ma che potrebbero dare come risultato un codice eseguibile non corretto. Dopo la ridirezione dell'output viene indicata una serie di otto opzioni, seguita dal nome del file da compilare. Vediamo in dettaglio il significato di ogni opzione.

-Q Fa sì che il compilatore memorizzi in un file di nome "AztecC.err" i messaggi relativi agli eventuali errori di compilazione. Dal momento che questo file non viene generato se non si verificano errori di compilazione, ce ne serviamo per rilevare se la compilazione ha avuto successo oppure no.

-n Ordina al compilatore di generare un modulo oggetto contenente anche le informazioni necessarie affinché il linker, se viene eseguito indicando l'opzione -g, generi il file .dbg oltre al file eseguibile. Il file .dbg serve se si impiega il *Source Level Debugger* per effettuare il debug del programma, ma serve anche al nostro file comandi per rilevare se la fase di link ha avuto successo oppure no.

+C Si specifica che, per quanto riguarda il codice eseguibile, il programma va compilato facendo uso del modello grande di memoria.

+D Si specifica che, per quanto riguarda i dati, il programma va compilato facendo uso del modello grande di memoria.

+L Forza il compilatore a considerare da 32 bit tutte le variabili di tipo int e le costanti dichiarate nel file sorgente.

+m Per tutte le funzioni standard del C che vengono chiamate nel sorgente, quest'opzione causa l'inserimento, all'inizio dei relativi codici, di una piccola routine che controlla la profondità dello stack; poiché questo controllo rallenta l'esecuzione e fa aumentare le dimensioni del program-

ma eseguibile, è bene ometterlo nella versione definitiva.

+p Garantisce la massima portabilità del programma eseguibile. Costringe infatti a utilizzare il modello di memoria grande sia per i codici eseguibili sia per i dati, forza a 32 bit le variabili di tipo int e garantisce l'integrità dei registri D2 e D3 del 68000 durante ogni chiamata di funzione. Non ci sono problemi nell'uso combinato di quest'opzione e delle +D, +C e +L, anche se in effetti indicando +p le opzioni +C e +D diventano superflue.

+r Informa il compilatore che è possibile utilizzare anche il registro D4 per memorizzarvi una variabile di tipo register. Ricordiamo che l'uso delle variabili register è spiegato nel manuale Manx.

Se al termine della compilazione è presente nella directory un file di nome AztecC.err, significa che durante la compilazione si sono verificati alcuni errori. In questo caso, come accade anche per il linker, viene avviato un nuovo processo in background che chiama l'editor per visualizzare il contenuto del file AztecC.err. Si noti che per la chiamata all'editor abbiamo usato l'alias ee, al quale ognuno può associare il nome del proprio editor preferito, come ED, MEMacs del *Workbench 1.3*, TxEd. Non usiamo un semplice comando TYPE in quanto il file AztecC.err potrebbe anche essere abbastanza lungo. Comunque, se si possiede la versione 1.3 del *Workbench* è possibile chiamare al posto dell'editor il tool MORE, utile per visualizzare a pagine il contenuto di un file. Si noti anche che abbiamo ridiretto l'output del comando RUN verso il dispositivo logico NIL:; un dispositivo fittizio che può ricevere qualunque file ma che non lo memorizza in nessun caso. Solitamente si ridirige l'output di un comando verso questo particolare dispositivo quando non è tanto utile da essere salvato in un file, e non si desidera neanche che venga visualizzato sullo schermo. Eseguito il comando RUN, il controllo viene ceduto all'etichetta END per concludere l'esecuzione del file comandi.

Se invece la compilazione ha avuto successo, alcuni prompt avvisano l'utente, e viene eseguito il comando FAILAT, il quale serve per stabilire il minimo codice d'errore che deve verificarsi perché venga interrotta l'esecuzione del file comandi. Questa soglia normalmente è 10, ma per tutta la fase di link ci conviene impostarla temporaneamente a un valore molto alto. Il motivo è che il linker, a differenza del compilatore, restituisce un codice d'errore ogni volta che si verifica una condizione d'errore, e tale codice è proprio un numero di errori che si sono verificati. Per fare un esempio, se non si provvede a indicare una particolare libreria linked della quale il nostro sorgente richiama 21 funzioni, durante la fase di link si verificano 21 errori "Undefined symbol: ...", e il linker restituisce il codice d'errore 21. È quindi facile intuire che se con un programma di

medie dimensioni si dimentica d'indicare nella fase di link una libreria linked particolarmente importante, il numero di errori che si verificano può diventare anche molto elevato. Impostando temporaneamente la soglia a 1000 ci assicuriamo un buon margine di sicurezza.

Prima di chiamare il linker, eseguiamo un controllo molto simile a quello compiuto per il file AztecC.err prima di chiamare il compilatore. Verifichiamo cioè se esiste nella directory corrente un file con lo stesso nome del file passato come argomento al file comandi, ma con l'estensione ".dbg", e se esiste lo cancelliamo. Infatti il linker genera questo file soltanto se la fase di link ha successo.

A questo punto, chiamiamo il linker, ridirigendo l'output verso il file RAM:ln.w. Nella chiamata indichiamo l'opzione -g perché venga creato il file ".dbg" richiesto dal *Source Level Debugger*, ma anche necessario al nostro file comandi per rilevare se la fase di link ha avuto successo. La seconda opzione che indichiamo, -t, serve per richiedere al linker di generare un file ASCII contenente tutti i simboli presenti nel programma e i loro indirizzi esadecimali relativi. Segue il nome del file oggetto da sottoporre alla fase di link, e le due librerie cl32.lib e sl32.lib, le più comuni nella programmazione.

Conclusa la fase di link, viene ripristinato il codice d'errore 10 come soglia oltre la quale viene interrotta l'esecuzione del file comandi, e viene visualizzato con l'editor il file RAM:ln.w qualora non sia presente su disco il file di estensione ".dbg", cioè qualora si sia verificato almeno un errore. Se invece tutto è andato bene, il file comandi visualizza alcuni prompt e termina all'etichetta END.

Per quanto sia già molto utile, il file comandi che abbiamo proposto può essere notevolmente migliorato, magari per compilare diversi moduli sorgente e sottoporli tutti alla stessa fase di link. Permette di risparmiare tempo, e in un sistema multitasking come l'Amiga consente di trasformare un file da sorgente a eseguibile mentre l'utente svolge altre mansioni. A questo proposito, osserviamo che se si desidera eseguire il file comandi in background, è opportuno eliminare tutti i comandi ECHO, così da non interferire con la finestra nella quale l'utente sta eseguendo altre operazioni.

Il prossimo articolo

Nel prossimo numero analizzeremo la versione 5.0 del compilatore C della Lattice, un prodotto molto interessante ed evoluto. Intendiamo così dare ai lettori la possibilità di confrontare tra loro i due migliori pacchetti di programmazione in C attualmente disponibili per l'Amiga.

Questa nuova release offre il supporto per tutti i microprocessori della Motorola, compreso il 68030 che il Manx ancora non riconosce; inoltre genera più in fretta il codice oggetto e quello

eseguiibile, codice tra l'altro meglio ottimizzato, ma soprattutto dispone di un editor dalle caratteristiche uniche, ossia con la possibilità di fermare il processo di compilazione a ogni errore per andare automaticamente a esaminare il file sorgente e correggerlo.

In questa versione è anche compreso un nuovo source level debugger, il *CodeProbe*, equivalente al *Source Level Debugger* della Manx, e un tool utile per ridurre lo spazio occupato dai file eseguibili risultanti dalla compilazione. Sono state anche apportate alcune migliorie ai vari tool già esistenti nelle versioni precedenti del pacchetto. Non mancano alcune nuove utility: per esempio un disassembler più completo e un tool che tiene sotto controllo il tempo di CPU utilizzato da ciascun programma in C, consentendo al programmatore di concentrare i suoi sforzi sull'ottimizzazione dell'efficienza del programma.

Listato 1: Startup-sequence

```
ECHO "Questo e' un file comandi di startup"

FAILAT 100

STACK 6000

ASSIGN a: DF0:
ASSIGN b: DF1:
ASSIGN r: RAM:
MAKEDIR RAM:c
MAKEDIR RAM:s
MAKEDIR RAM:t

PATH add RAM:c RAM: DF0:c DF0: DF1:c DF1:

COPY a:c/TxEd ric:ee
COPY a:c/copy ric
COPY a:c/delete ric
COPY a:c/list ric
COPY a:c/grep ric

; Assegnazione delle variabili globali per il
; sistema Manx

SET ZOPT=RAM:z:opt
SET TAGS=RAM:global:tags
SET FUNCLIST=RAM:manx:c
SET INCLUDE=DF0:include
SET CTEMP=RAM:t
SET CLIB=DF0:lib:DF1:lib
SET DBINIT=DF0:s:dbinit

; Attiva le estensioni di TxEd

FASTFONTS
BLITZDISK

; Attiva WShell e il handler ConMan

SETEKEXECUTE
CONMAN -c
```

NEWSH

```
MAKEDIR RAM:env
ASSIGN ENV: RAM:env
COPY DF0:s/env ENV:

; Definisce gli alias. Si noti che
; se si dispone solo dello Shell
; occorre inserire questi alias nel
; file $Shell-Startup, e sostituire
; il carattere "=" con uno spazio. Inoltre,
; non tutti i comandi
; elencati sono disponibili nella versione 1.3
; del Workbench

ALIAS 0=cd DF0:
ALIAS 1=cd DF1:
ALIAS 2=cd RAM:
ALIAS a=assign
ALIAS aa=alias
ALIAS ab=addbuffers
ALIAS ar=arun
ALIAS bd=binddrivers
ALIAS bz=blitzdisk
ALIAS c=copy
ALIAS cm=conman
ALIAS ctp=changetaskpri
ALIAS d=dir
ALIAS d0=dir DF0:
ALIAS d1=dir DF1:
ALIAS d2=dir RAM:
ALIAS da=dir DF0:
ALIAS dac=dir DF0:c
ALIAS dad=dir DF0:devs
ALIAS daf=dir DF0:fonts
ALIAS dai=dir DF0:include
ALIAS dal=dir DF0:l
ALIAS dalib=dir DF0:lib
ALIAS dalibs=dir DF0:libs
ALIAS dblibs=dir DF1:libs
ALIAS daf=dir > RAM:dira DF0: opt a
ALIAS dbf=dir > RAM:dirb DF1: opt a
ALIAS dap=dir > PAR: DF0: opt a
ALIAS dbp=dir > PAR: DF1: opt a
ALIAS das=dir DF0:s
ALIAS dat=dir DF0:t
ALIAS db=dir DF1:
ALIAS dbc=dir DF1:c
ALIAS dd=diskdoctor
ALIAS dg=diskchange
ALIAS dl=delete
ALIAS dlibs=dir DF0:libs
ALIAS dr=dir RAM:
ALIAS drc=dir RAM:c
ALIAS e=execute
ALIAS ea=ee DF0:
ALIAS eb=ee DF1:
ALIAS ec=ee DF0:s/cccln
ALIAS ee=TxEd
ALIAS ess=ee DF0:s/startup-sequence
ALIAS ff=fastfonts
ALIAS fk=funkeys
ALIAS fn=filenote
ALIAS fm=format
ALIAS fv=fileviewer
ALIAS g=grep
ALIAS ga=cd DF0:
ALIAS gb=cd DF1:
ALIAS gas=cd DF0:s
ALIAS gbs=cd DF1:s
ALIAS h=pushed
```

```

ALIAS hh=history
ALIAS is=install
ALIAS i=info
ALIAS j=join
ALIAS l=list
ALIAS ll=loadlib
ALIAS md=mkdir
ALIAS mm=mount
ALIAS mt=mounted
ALIAS nc=newcli
ALIAS ns=newwsh
ALIAS p=path
ALIAS pp=prompt
ALIAS pc=popcli
ALIAS pcd=prompt %c
ALIAS pet=prompt %e
ALIAS pi=prompt %i
ALIAS p2=prompt %2
ALIAS p3=prompt %3
ALIAS p4=prompt %4
ALIAS o=popcd
ALIAS pt=protect
ALIAS r=run
ALIAS re=resident
ALIAS rl=rlabel
ALIAS rm=rexxmast
ALIAS rn=rename
ALIAS rr=resi
ALIAS rw=rwsh
ALIAS s=swpcd
ALIAS sd=setdate
ALIAS sh=search
ALIAS sm=setmap
ALIAS se=setexecute
ALIAS scm=setcman
ALIAS sk=stack
ALIAS ss=status
ALIAS stp=settaskpri
ALIAS t=type
ALIAS vv=version
ALIAS wtp=waitforport
ALIAS w=wait
ALIAS x=rexx
ALIAS xl=rllib
ALIAS xs=risset

```

```

; Trasforma in residenti alcuni comandi.
; Si noti che se si dispone solo
; dello Shell occorre cambiare RESI in RESIDENT,
; e che il comando ALIAS
; e' gia' residente

```

```

RESI DF0:c/alias
RESI DF0:c/connan
RESI DF0:c/cdtitle
RESI DF0:c/history
RESI DF0:c/newwsh
RESI DF0:c/resi
RESI DF0:c/runwsh
RESI DF0:c/cd
RESI DF0:c/dir
RESI DF0:c/info
RESI DF0:c/mkdir
RESI DF0:c/path
RESI DF0:c/run

```

```
: Legge in memoria il file ProgrammingHistory
```

```
history DF0:ProgrammingHistory
```

```

; Premere CTRL-B per cancellare il buffer
; Attiva l'interprete residente dell'AREXX e
; gli assegna il dispositivo logico REXX:

REXXMAST

ASSIGN REXX: DF0:rexx

; Chiude la finestra del CLI

ENDCLI > NIL:

```

Listato 2: File comandi ccln

```

;KEY file

IF "<file>" EQ ""
ECHO "Indicare il nome del sorgente dopo ccln"
SKIP END
ENDIF

IF NOT EXISTS <file>.c
ECHO "Il file <file>.c non esiste"
SKIP END
ENDIF

IF EXISTS Aztec0.err
DELETE Aztec0.err
ENDIF
ECHO "Compilo e assemblo il file"

cc > RAM:cc.w -Q -n +C +D +L +m +p +r <file>

IF EXISTS Aztec0.err
ECHO "Errori di compilazione"
RUN > NIL: ee Aztec0.err
SKIP END
ENDIF

ECHO ""
ECHO "Compilazione conclusa con successo"
ECHO "Esaminare il file RAM cc.w per i warning"
ECHO "Procedo con la fase di link"

FAILAT 1000

IF EXISTS <file>.dbg
DELETE <file>.dbg
ENDIF

ln > RAM:ln.w -g -t <file> s132.lib c132.lib

FAILAT 10

IF NOT EXISTS <file>.dbg
ECHO "Errori nella fase di link"
RUN > NIL: ee RAM:ln.w
SKIP END
ENDIF

ECHO ""
ECHO "Fase di link conclusa con successo"
ECHO "Esaminare il file RAM ln.w per i warning"

LAB END

```

Amiga 2000

Amiga 2000 è il collaboratore ideale per preparare facilmente ed in pochissimo tempo al video testi e grafici e riportarli poi su carta, su lucidi o diapositive. È uno strumento creativo semplice e immediato per realizzare soggetti fantastici in due o tre dimensioni con 4.096 colori. Qualunque sia la tua professione, con Amiga 2000 hai a disposizione le meraviglie dell'ambiente Amiga DOS e del sistema MS-DOS, con i quali potrai elaborare i tuoi progetti con una grafica ineguagliabile. In più, Amiga 2000 è Commodore: un marchio leader nel mondo degli home computer e dei sistemi professionali. Non a caso il primo PC (P.E.T.) è stato prodotto da Commodore, che può così vantare la più lunga esperienza nel settore.

- MS-DOS e XENIX sono marchi registrati della Microsoft Inc.
- UNIX è un marchio registrato della AT&T.
- Commodore e Amiga sono marchi registrati della Commodore Inc.



Amiga 2000.

Per te
che vuoi
un amico
sulla
scrivania.



Amiga 2000.

Per te
che vuoi
lavorare
con un creativo
alla tua altezza.



Amiga 2000.

Per te e per
i tuoi progetti
che non hanno
bisogno solo
del sistema
MS-DOS.



C'è un Commodore p

PC 10.

Per te
che vuoi
un personal
che faccia
scuola.

PC 40.

Per te
che vuoi lavorare
solo con
i numeri uno.

PC 60.

Per te
che vuoi
una gamma
completa
di soluzioni
al servizio
della tua
Azienda.



La linea PC Commodore garantisce nel mondo MS-DOS - grazie ai modelli XT, AT e 386 - la soluzione ad ogni problema. Il PC10 (XT) e PC20 (XT) rappresentano la soluzione ideale di partenza. I modelli PC40 (AT) assicurano la potenza e l'espandibilità necessarie a soddisfare ogni esigenza di crescita in qualunque settore. Il PC60 (386) - grazie alla sua potente architettura - è il top della gamma. Sui modelli PC40 e PC60 potrai installare anche il sistema operativo Xenix® o Unix® se la tua azienda lo richiede, e potrai inoltre collegarli in rete. In più PC10, 20, 40, 60 sono Commodore: un marchio leader nel mondo degli home computer e dei sistemi professionali. Non a caso il primo PC (P.E.T.) è stato prodotto dalla Commodore, che può così vantare la più lunga esperienza nel settore.

Per informazioni, dalle 14 alle 17.30



er te, chiunque tu sia.



Lombardia

Milano

- A'ION - Via Bigli, 11
- AL RISPARMIO - V.le Monza, 204
- B.C.S. - Via Montegani, 11
- BRAHA ALBERTO - Via Pier Capponi, 5
- EDI CARED - Via Pietro Calvi, 20
- E.D.S. - C.so Porta Ticinese, 4
- FLOPPERIA - V.le Montenero, 31
- GI-SETTANTA - Via Burlamacchi, 4
- GIGLIONI - V.le Luigi Sturzo, 45
- LOGITEK - Via Golgi, 60
- MARCUCCI - Via F.lli Bronzetti, 37
- NEWEL - Via Mac Mahon, 75
- SUPERGAMES - Via Vitruvio, 38
- 68000 E DINTORNI - Via Washington, 91

Provincia di Milano

- ALL COMPUTER - Residenza Sassi, 312 - Milano 3 - Basiglio
- PENATI - Via Verdi, 28/30 - Corbetta
- S.A.T. - Via Milano, 24 - Nerviano
- IL CORSOIRE - Via Campo dei Fiori, 35 - Novate Milanese

Bergamo

- DERCOM - Via Borgo Palazzo, 65/A

Provincia di Bergamo

- COMPUTER SHOP - Via V. Veneto, 9 - Capriate San Gervasio

Brescia

- MASTER INFORMATICA - Via F.lli Ugoni, 10/B

Provincia di Brescia

- CAVALLI PIETRO - Via X Giornate, 14/B - Castrezzato
- DATA SYSTEM NEW - Via Gramsci, 33 - Concesio
- MEGABYTE - P.zza Maluzzi, 14 - Desenzano del Garda

Provincia di Como

- CIMA ELETTRONICA - Via Leonardo da Vinci, 7 - Lecco
- FUMAGALLI - Via Cairoli, 48 - Lecco
- RIGHI ELETTRONICA - Via Leopardi, 26 - Olgiate Comasco

Cremona

- MONDO COMPUTER - Via Giuseppina, 11/B
- PRISMA - Via Buoso da Dovara, 8

Provincia di Cremona

- EUROELETTRONICA - Via XX Settembre, 92/A - Crema

Provincia di Mantova

- CLICK - ON COMPUTER - S.S. Goietese, 168 - Goito

Pavia

- POLIWARE - C.so Carlo Alberto, 76

Provincia di Pavia

- LOGICA MAINT - Via Montegrappa, 32 - Vigevano

Provincia di Sondrio

- FOTONOVA - Via Valeriana, 1 - San Pietro di Berbenno

Varese

- IL CENTRO ELETTRONICO - Via Morazzo, 2
- SUPERGAMES - Via Carrobbio, 13

Provincia di Varese

- CURIO TRE - Via Ronchetti, 71 - Cavaria
- J.A.C. NUOVE TECNOLOGIE - C.so Matteotti, 38 - Sesto Calende

Piemonte

Cuneo

- ROSSI COMPUTERS - C.so Nizza, 42

Provincia di Cuneo

- PUNTO BIT - C.so Langhe, 26/C - Alba

Novara

- SOFTTEAM - Via Locchi, 6

Provincia di Novara

- L.A.E. SOFTWARE - C.so Cavour, 46/59 - Arona
- ALL COMPUTER - C.so Garibaldi, 106 - Borgomano - ro
- ELLIOTT COMPUTERSHOP - Via Don Minzoni, 32 - Intra

Torino

- ALEX COMPUTERS - C.so Francia, 233/4
- DE BUG COMPUTER - C.so Vittorio Emanuele II, 22
- DESME UNIVERSAL - Via San Secondo, 95
- IL COMPUTER - Via Nicola Fabrizi, 126
- STAF - C.so Regina Margherita, 97

Provincia di Torino

- DIAM INFORMATICA - C.so Francia, 146 BIS - Cascine Vica - Rivoli

Provincia di Vercelli

- C.S.I. TEOREMA - Via Losana, 9 - Biella

- CHIP - Via Martiri della Libertà, 47 - Cosato

Veneto

Belluno

- UP TO DATE - Via Vittorio Veneto, 43

Padova

- SARTO COMPUTER - Via Armistizio, 79

Trentino Alto Adige

Bolzano

- COMPUTER POINT - Via Roma, 82

Provincia di Bolzano

- ELEKTRO TAPPEINER - P.zza Principale 90 - Silandro

Trento

- CRONST - Via G. Galliei, 25

Friuli Venezia Giulia

Pordenone

- SIEL - Via Colonna, 45

Udine

- CO.REL. ITALIANA - Via Tavagnacco, 91
- MOFERT 2 - Via Leopardi, 21

Liguria

Genova

- ODEL - Via Orsini, 4 R

- SALS INFORMATICA - Via G. D'Annunzio, 2

Emilia

Bologna

- MINNELLA ALTA FELDTL - Via Mazzini, 146/2
- SINIOPIA INFORMATICA - V.le Pietramellara, 11

Provincia di Bologna

- S.C. COMPUTERS - Via Enrico Fermi, 4 - Castel San Pietro
- S.P.E. INFORMATICA - Via di Mezzo Ponente, 385 - Crevalcore

Modena

- VIDEO VAL WILLY COMPUTERS - Via Ca-naletto, 223

Provincia di Modena

- NEW MEDIA SYSTEM - Via Roma, 281 - Solleria

Parma

- NEW LIST COMPUTER - Via Nazario Sauro, 9

Reggio Emilia

- COMPUTERLINE - Via San Rocco, 10/C
- POOL SHOP - Via Emilia S. Stefano, 9/C

Romagna

Repubblica di San Marino

- A.C.S. - Via Nonagualdaria - Caiungo

Ferrara

- BUSINESS POINT - Via Carlo Mayer, 85

Provincia di Forlì

- TOP BIT - Via Veneto, 12 - Forlimpopoli
- EASY COMPUTER - Via Lagomaggio, 50 - Rimini
- NUMERO - Via Battaglini, 21 - Rimini

Provincia di Ravenna

- E.T.S. - Via Saffi, 1 - Alfonsine
- P.L.Z. INFORMATICA - P.zza Sercognani, 6 - Faenza

Toscana

Arezzo

- DELTA SYSTEM - Via Piave, 13

Firenze

- M.T.S. DISTRIBUZIONE - Via di Novoli, 64 A
- TELEINFORMATICA TOSCANA - Via Bronzino, 36

Grosseto

- COMPUTER SERVICE - Via Dell'Unione, 2

Livorno

- ETA BETA COMPUTER - Via S. Francesco, 30
- FUTURA 2 - Via Cambini, 19

Provincia di Lucca

- IL COMPUTER - V.le Colombo, 216 - Lido di Camaione

Pisa

- ELECTRONIC SERVICE - Via della Vecchia Tranvia, 10

Pistoia

- ELECTRONIC SHOP - Via degli Scalzi, 3

Siena

- RENATO BROGI - P.zza Gramsci, 28

Provincia di Siena

- ELETTRONICA di BIFOLCHI - Via di Gracciano nel Corso, 111 - Montepulciano
- ELETTROMERCATO di BURRINI - Via Toscana, 6 - Monteriggioni

Umbria

Perugia

- MIGLIORATI - Via S. Ercolano, 3
- STUDIO SYSTEM - Via R. D'Andreotto, 49/51

Lazio

Roma

- D.R.R. - Via Giovanni Giorgi, 6

Abruzzo

Teramo

- MAX CENTER - V.le Crispi, 68

Puglia

Bari

- ARTEL - Via Guido d'Orso, 9
- COMPUTER'S ARTS - V.le Meucci, 12/B
- ELETTRONICA SISTEMI - V.le della Repubblica, 67/69
- PALLICELLI SABINO & FIGLI - Via Fanelli, 231/C

Campania

Napoli

- DARVIN - Calata San Marco, 25
- DPC INFORMATICA - Via E. Nicolardi, 129
- GENERAL COMPUTERS - Via Bernini, 101
- SPY - Via Domenico Fontana, 135
- TOP VIDEO - TOP COMPUTER - Via S. Anna dei Lombardi, 12

Provincia di Napoli

- ITALIANA SOFTWARE - Via Zara, 11 - Acerra
- NUOVA INFORMATICA SHOP - Via Libertà, 185 - Portici

Provincia di Salerno

- COMPUTER WORLD - Via Quarto, 6 - San Marzano sul Sarno

Calabria

Cosenza

- SIRANGELO COMPUTER - Via N. Parisio, 25

Reggio Calabria

- CONTROL SYSTEM - Via S. Francesco da Paola, 49/D-E
- SYSTEM HOUSE - Via Fiume ang. Palestri-no, 1

Provincia di Reggio Calabria

- COMPUTER SHOP - Via Matteotti, 48/50 - Locri

Sicilia

Provincia di Enna

- ITALSOFT - Via Dottor Palazzolo - Agrig

Palermo

- C.H.P. - Via Libertà, 95

cambia in  Commodore®

Esigi sempre la garanzia della Commodore Italiana S.p.A.

CYGNUSED PROFESSIONAL, UN EDITOR PER I PROGRAMMATORI

Un prezioso aiuto per la programmazione: l'editor CygnusEd Professional. Le sue caratteristiche più significative sono la gestione su varie finestre di uno o più file di testo e la compatibilità AREXX

di Eugene P. Mortimore

Senz'altro vi sarà capitato di desiderare un editor di testi che consenta l'apertura di dieci finestre su dieci file diversi, oppure che permetta la visione delle diverse parti di un file in varie finestre, facilitando così le operazioni di cut & paste. Magari vi piacerebbe anche che queste finestre venissero ingrandite automaticamente a tutto schermo quando le selezioniamo, e che la configurazione e i comandi dell'editor potessero essere personalizzati in modo completo. E se ci fosse anche la possibilità di recuperare i file che dopo un crash del sistema diamo per spacciati, saremmo davvero di fronte all'editor ideale.

Ma questo editor esiste davvero: la ASDG Incorporated ha lanciato sul mercato *CygnusEd Professional*, un programma che congloba tutte le caratteristiche elencate e molte altre, come la possibilità di controllare l'editor attraverso i linguaggi di programmazione più famosi, come C, Basic o Pascal, per scavalcare la consueta interazione con il mouse e la tastiera, e la presenza di un linguaggio semplificato ma potente per definire macro e routine. Si tratta di uno dei più importanti programmi per l'Amiga e merita tutta la nostra attenzione se siamo assidui utenti di editor e word processor. In aggiunta alle caratteristiche dei migliori word processor, *CygnusEd Professional* offre inoltre una serie di opzioni che lo rendono particolarmente adatto alla scrittura di programmi. Dopo aver preso confidenza con queste possibilità, scrivere sorgenti diventerà enormemente più rapido e facile.

Aspetti generali

Esaminiamo ora le caratteristiche più notevoli di questo pacchetto. Articoleremo l'analisi in sette punti principali:

- la creazione di finestre, in numero massimo di dieci
- l'assegnazione di viste logiche a file diversi o parti dello stesso file
- la semplificazione dei passi da compiere per selezionare blocchi di testo e per memorizzare sequenze di comandi sotto forma di macro
- l'esecuzione di comandi AmigaDOS e di file comandi AREXX senza uscire dall'editor
- il recupero istantaneo dei file presenti in memoria quando avviene un crash del sistema
- la possibilità di rendere residente l'editor e di richiamarlo attraverso una sequenza di tasti (hot key) in qualunque momento
- la possibilità di ridefinire i parametri globali dell'editor a proprio piacimento.

Apertura di più viste

Scrivere un programma richiede quasi sempre l'uso di più file sorgente. Per esempio, in un programma in C capita spesso di avere un file contenente soltanto la funzione main(), e uno o più file dedicati alle sotto-funzioni da questa richiamate. L'editor *CygnusEd* consente di tenere aperti contemporaneamente tutti questi file in opportune finestre, fino a un massimo di dieci, e di saltare con il cursore dall'una all'altra con il solo uso del mouse, ingrandendo tra l'altro in maniera automatica la finestra di volta in volta selezionata, fino a farle occupare le dimensioni dell'intero schermo. Facciamo subito un esempio d'impiego pratico: l'utente può tenere lo schermo permanentemente suddiviso in due finestre, collocando in quella superiore la funzione main(), e in quella inferiore gli altri sorgenti a cui la funzione main() fa di volta in volta riferimento.

Chiariamo subito che queste finestre non sono gestite da Intuition e quindi non si possono spostare, ridimensionare o sovrapporre facendo uso del mouse. Sono in pratica semplici suddivisioni orizzontali dello schermo, sulle quali possiamo agire soltanto attraverso i comandi previsti dall'editor. D'altra parte, non occorre quasi mai sovrapporre finestre contenenti testi, e diminuire le loro dimensioni orizzontali serve soltanto a rendere i file più difficili da leggere; inoltre, affidare a *CygnusEd* anziché a Intuition la gestione dello spazio disponibile sullo schermo permette di evitare la presenza di bordi e barre di spostamento aumentando così lo spazio disponibile.

Ora, supponiamo che il nostro programma sia composto da un file sorgente contenente la funzione `main()`, e da un secondo file contenente cinque funzioni di supporto. Conviene dedicare una finestra alla sola funzione `main()`, e visualizzare in altre cinque ognuna delle funzioni contenute nel secondo file sorgente. Evidentemente, poiché queste cinque funzioni fanno parte di un unico file, tutte le modifiche introdotte nelle loro finestre si ripercuotono sullo stesso file.

In generale, poter vedere su finestre diverse più parti dello stesso file è sempre utile durante la programmazione: permette di risparmiare tempo e soprattutto di mantenere sempre una visione globale del lavoro. E soprattutto, permette d'identificare subito i blocchi di testo da copiare o da spostare da una finestra all'altra.

Per una visione schematica delle operazioni consentite da *CygnusEd* nella gestione delle finestre, si consulti la Tavola 1, che mostra tutte le sue più importanti caratteristiche.

Le operazioni sui blocchi di testo e le macro

CygnusEd utilizza il dispositivo Clipboard dell'Amiga per le operazioni di cut & paste, e offre una vasta gamma di scelte per quanto riguarda lo spostamento di blocchi di testo tra finestre diverse. Si possono definire due tipi di blocchi: quello tradizionale "a righe" oppure quello "rettangolare", ossia da colonna a colonna e da riga a riga, che è utile soprattutto per le tabelle. Le varie possibilità sono meglio evidenziate nella sezione dedicata al menu Cut/Paste della Tavola 1.

Si possono poi definire macrocomandi, cioè sequenze di comandi avviabili con un'unica istruzione. I macrocomandi si rivelano utili per automatizzare sequenze di operazioni a cui si fa frequente ricorso. Una particolarità interessante, peraltro comune anche all'editor *MEMacs* distribuito con la versione 1.3 del *Workbench*, è che questi macrocomandi vengono creati semplicemente eseguendo in sequenza le operazioni desiderate dopo aver abilitato il modo di registrazione. Conclusa la registrazione, si può salvare su disco il macrocomando ottenuto e mandarlo in esecuzione non appena si ripresenterà la necessità della stessa sequenza di operazioni.

L'interfaccia con il pacchetto AREXX

Il sistema *AREXX* consente di stabilire canali di comunicazione fra le applicazioni dotate di opportuna interfaccia, oppure di sostituire la consueta interazione utente-applicazione (tramite tastiera o mouse) con file batch di comandi (al sistema *AREXX* abbiamo dedicato parte dell'articolo "*AREXX* e *WShell*", due utility per il programmatore" in questo stesso numero di *Commodore Gazette*).

CygnusEd, come abbiamo già detto, è un'applicazione dotata d'interfaccia *AREXX* (anzi, è una delle prime apparse sul mercato). Questo significa che è in grado di ricevere input non solo dalla finestra di cui dispone sullo schermo, ma anche dalla message port *AREXX* che crea in fase d'inizializzazione e che rende pubblica nel sistema. L'interfaccia *AREXX* prevede lo scambio di messaggi costruiti secondo un particolare formato al quale devono uniformarsi tutte le applicazioni *AREXX* compatibili. Generalmente, i messaggi che passano attraverso queste message port contengono comandi e dati.

Nel caso di *CygnusEd*, la sua interfaccia *AREXX* ci consente di ricorrere a file comandi *AREXX* per attivare molte funzioni che normalmente selezioneremmo tramite i menu, e permette inoltre di svolgere particolari compiti non previsti dall'editor, come comandare altre applicazioni *AREXX* compatibili senza uscire da *CygnusEd*. Come si può vedere, la compatibilità *AREXX* apre nuove possibilità, superiori perfino a quelle fornite dall'utilissima gestione delle macro.

Molto meno sensazionale, ma pur sempre utile, è la possibilità d'impartire comandi all'AmigaDOS direttamente dall'editor, senza dover aprire una finestra CLI. Il canale di output standard di questi comandi diventa una delle finestre dell'editor.

I parametri globali per la personalizzazione dell'editor

Si supponga di essere impegnati contemporaneamente in diverse attività, come la scrittura di un libro, di alcune lettere e di un programma in Basic: il proprio editor dovrebbe lavorare come un word processor per la stesura del libro e delle lettere, e dovrebbe invece configurarsi come un editor adatto alla stesura di programmi quando si cambia contesto. La configurazione per il funzionamento da editor di sorgenti prevederebbe per esempio particolari tabulatori, nonché funzioni speciali quali il controllo automatico sulla chiusura delle parentesi, mentre la configurazione per il funzionamento da word processor prevederebbe piuttosto l'attivazione di particolari margini e allineamenti per la formattazione dei testi.

Con *CygnusEd* è possibile crearsi alcune configurazioni personalizzate impostando opportunamente i parametri ai quali il programma accede per la gestione delle proprie risorse. La configurazione attiva può essere salvata e caricata in qualunque momento.

I menu di CygnusEd

Il modo migliore per rendersi conto della varietà di scelte messa a disposizione da *CygnusEd Professional* è dare uno sguardo alle voci dei suoi menu, che spesso mostrano caratteristiche inedite. Nella Tavola 1 sono riassunte molte di queste voci, in particolare quelle che sono tipiche di *CygnusEd*. Le suddivisioni seguono quelle degli otto sotto-menu presenti sulla barra menu principale. La maggior parte delle opzioni disponibili si possono attivare anche attraverso combinazioni di tasti CTRL o ALT: non appena si diventa pratici del sistema, è possibile accelerare le operazioni facendo uso della tastiera anziché dei menu a discesa.

Nel caso si desideri comandare l'editor attraverso la sua interfaccia *AREXX*, la sintassi dei comandi da inviare corrisponde alle stringhe di caratteri presenti nelle stesse voci dei menu.

Tutte le voci dei menu che terminano con i puntini di sospensione, come per esempio "Include file...", necessitano di ulteriori dati da parte

dell'utente, tanto che la selezione avvenga interattivamente quanto che si ricorra a un file batch *AREXX*.

Il menu Project – Il menu Project mette subito in evidenza alcune opzioni insolite per un editor. È possibile per esempio cancellare il file presente all'interno di una finestra, lasciando attiva la finestra; si può inserire un file all'interno di un altro file con il comando "Include"; si può salvare su disco una parte di testo compresa tra due righe (il consueto blocco di testo), oppure un'area rettangolare compresa non solo fra due righe ma anche fra due colonne (il cosiddetto "blocco rettangolare"); si può cambiare la directory corrente dell'AmigaDOS senza uscire dall'editor; si può stampare un blocco di testo, oppure indirizzare la stampa verso un file su disco; infine esiste la possibilità di uscire dall'editor, disabilitando anche la hot key di riattivazione.

Il menu Environment – **Environment** è la prima parte di un menu composto da tre sezioni, la prima delle quali evidenziata dalla lettera "E" maiuscola.

TAVOLA 1

Comando

Descrizione e impiego

Menu Project

Clear	Cancella il file corrente dalla memoria, senza chiudere le finestre aperte
Include file...	Inserisce un file nel testo corrente alla posizione indicata dal cursore
Save Block...	Salva in un file il blocco di testo selezionato
Save All Changes	Salva tutti i file che hanno subito modifiche dopo l'ultimo salvataggio
Change Current Directory	Cambia la directory corrente dell'AmigaDOS
Print Block...	Invia il contenuto del blocco a una stampante o a un file
Print File...	Invia il file a una stampante o a un file
Quit & Die	Chiude <i>CygnusEd</i> completamente, cioè disabilitando anche l'hot key che lo riattiva istantaneamente

Menu Environment

Screen height	Assegna allo schermo un'altezza tra 200 e 236 pixel o attiva l'interlacci
Screen width	Imposta la larghezza dello schermo a un valore tra 640 e 704 pixel
Force custom screen	Costringe l'editor a utilizzare uno schermo diverso da quello del <i>Workbench</i>
Macro definitions	Definisce una macro e la assegna a un tasto perché si possa mandare facilmente in esecuzione
Priority	Modifica la priorità che l'Exec riconosce al task <i>CygnusEd</i>
Autosave	Attiva il salvataggio di sicurezza periodico dei file
Set Icon Tool Name	Permette di variare il contenuto dell'opzione Default Tool di un'icona, ossia il programma che viene eseguito quando l'utente la seleziona due volte con il mouse
Change Colors	Seleziona una delle 24 possibili combinazioni di colori per lo schermo dell'editor
Do Safe Saves	Utilizza un particolare sistema per il salvataggio dei file, che usa più file temporanei per motivi di affidabilità
Icon Creation	Crea un'icona per ogni file di testo che viene salvato
HOT-Start Enabled?	Mantiene in memoria una versione residente di <i>CygnusEd</i> , per evitarne successivi caricamenti da disco
Auto-expand views?	Estende automaticamente la finestra attiva a tutto lo schermo
Keypad = movement?	Rende possibile spostare il puntatore del mouse attraverso la tastiera numerica
Load environment	Carica in memoria un file contenente i valori dei parametri globali e li rende attivi
Save environment...	Memorizza il valore dei parametri globali in un file

Menu eNVIronment

Tab size	Modifica le dimensioni dei tabulatori tra 1 e 10 colonne, e aggiorna il file corrente
Customize tabs	Scavalca il valore di default per i tabulatori, consentendo di specificarne uno per ogni riga
Set right border	Imposta il bordo destro del documento (utilizzato per il wordwrap e la formattazione)
Set scroll jump	Imposta la velocità di scoll
Layout?	Consente al cursore di muoversi oltre la fine di una riga, nello spazio non appartenente al file
Word wrap?	Rende automatico il wordwrap, in modo che non si debba premere CR a fine riga
Insert mode?	Passa dal modo inserimento alla sovrapposizione
Tab = spaces?	Trasforma le tabulazioni in serie di spazi

Menu environMENT

Status line	Visualizza o cancella la linea di stato nella barra titolo della finestra
White spaces	Serve a visualizzare come simboli speciali i caratteri non stampabili
Scroll bar	Visualizza o cancella la barra di scoll a destra o a sinistra dello schermo
Set scroll borders	Definisce l'area interessata dallo scoll all'interno della finestra

SEGUE

SEGUE TAVOLA 1

Menu Special

View operations	Permette di eseguire diverse operazioni sulle finestre
Format	Esegue la formattazione del testo in maniera che nessuna parola superi il margine destro impostato
DOS/AREXX interface	Serve a mettere in colloquio <i>CygnusEd</i> con l'AmigaDOS e con altre applicazioni <i>AREXX</i> compatibili
Enter ascii...	Permette d'inserire il codice ASCII di un carattere
Center cursor	Fa scorrere il testo nella finestra attiva in modo che la linea corrente risulti al centro della finestra
Center line	Centra la linea corrente tra i margini
Repeat key/menu...	Ripete la pressione di un tasto o la selezione di un menu un certo numero di volte
Find matching bracket	Ricerca la parentesi di chiusura relativa a quella su cui si trova il cursore
Mark location	Definisce fino a 10 posizioni all'interno del file, permettendo di spostarsi con facilità

Menu Cut/Paste

Mark block	Segnala l'inizio o la fine di un blocco di testo per successive operazioni di cut & paste
Mark columnar block	Analogo al precedente, ma si riferisce a blocchi rettangolari
Cut block	Esegue il cut, ossia sposta la parte di testo selezionata nel buffer Paste
Copy block	Copia nel buffer Paste la parte di testo selezionata
Insert block	Inserisce il contenuto del buffer Paste nella posizione indicata dal cursore
Save block to file...	Memorizza il blocco selezionato in un file
Rot block	Crittografa il testo (utile per trasmettere dati su network)
Strip CR block	Elimina i CR da un blocco di testo
Change case block	Inverte da maiuscolo a minuscolo tutti i caratteri di un certo blocco
Delete word	Cancella la parola a destra o a sinistra del cursore
Undelete word	Recupera le parole cancellate con la funzione precedente
Delete line	Cancella la riga su cui si trova il cursore
Delete to EOL	Cancella la riga corrente a partire dalla posizione del cursore, fino al fondo
Undelete line	Recupera le righe cancellate

Menu Search/Replace

Change case letter	Inverte da maiuscola a minuscola o viceversa la lettera su cui si trova il cursore
Change case word	Inverte da maiuscola a minuscola o viceversa l'intera parola su cui si trova il cursore
Upper case word	Trasforma in caratteri maiuscoli la parola su cui si trova il cursore
Lower case word	Opposto al precedente

Menu Move

Jump to line...	Muove il cursore su una particolare linea del testo
Jump to auto-mark	Muove il cursore su uno dei "mark" impostati
Begin of screen	Muove il cursore all'inizio dello schermo
End of screen	Muove il cursore alla fine dello schermo
Begin of line	Muove il cursore all'inizio della linea
End of line	Muove il cursore alla fine della linea
Up 12 lines	Muove il cursore di 12 linee verso l'alto
Down 12 lines	Muove il cursore di 12 linee verso il basso
Left 12 characters	Sposta il cursore verso sinistra di 12 caratteri
Right 12 characters	Sposta il cursore verso destra di 12 caratteri
Begin of file	Muove il cursore all'inizio del file
End of file	Muove il cursore alla fine del file
Prev word	Muove il cursore alla parola precedente
Next word	Muove il cursore alla parola successiva

Le sue voci più interessanti sono la definizione delle dimensioni dello schermo, l'impiego da parte di *CygnusEd* di un proprio schermo sul quale le operazioni vengono svolte più velocemente che sullo schermo del *Workbench* (probabilmente aprendo uno schermo custom dotato di un minor numero di bit plane e quindi di meno colori), la creazione delle macro e la possibilità di variare il livello di priorità dell'editor nel sistema multi-tasking dell'Amiga. L'opzione Set Icon Tool Name interviene quando un'icona viene selezionata due volte dal *Workbench* mentre Autosave attiva il salvataggio automatico dei file a intervalli regolari.

Questo menu permette anche di variare i colori dello schermo in qualsiasi istante e di scegliere il metodo che si ritiene più sicuro per effettuare le copie di backup dei propri file. Inoltre, si può scegliere di conservare una copia residente dell'e-

ditor in memoria, richiamabile con una hot key; si può fare in modo che la finestra attiva venga sempre estesa a tutto lo schermo; si può comandare il puntatore attraverso la tastierina numerica. Infine, tutti i parametri che descrivono la configurazione dell'editor possono essere salvati in un apposito file, in modo che - ricaricandolo in memoria - l'editor si riconfiguri esattamente com'era.

eNVIROnment è la seconda parte del menu: in essa sono presenti voci che permettono d'impostare i tabulatori, i margini, il wordwrap, la velocità di scroll; si può passare dalla modalità "wordprocessing", nella quale le parole vengono mandate a capo automaticamente al fondo di ogni riga, alla modalità "command-line", in cui ogni riga di testo è un'entità a sé stante, come nei primi editor di linea, nei quali era obbligatorio dare il comando di fine riga durante la digitazione, e per la correzione

si doveva segnalare esplicitamente la riga su cui si doveva intervenire. È infine possibile decidere se i tabulatori devono essere il carattere ASCII di tabulazione o una serie di spazi (un'opzione molto utile quando i file che si scrivono devono poi essere elaborati da altri programmi, che evidentemente non possono sapere a quale distanza corrisponde il tabulatore impostato all'interno di *CygnusEd*).

environment è la terza e ultima parte del menu. Ci offre la possibilità di abilitare o disabilitare la visualizzazione di una serie d'informazioni nella barra titolo che riguardano lo stato dell'editor; si possono inoltre visualizzare attraverso speciali codici video certi caratteri non stampabili come i tabulatori e i CR. Infine possiamo decidere se vogliamo le barre di scroll sulla finestra principale dell'editor.

Come si vede, il menu Environment, nel suo complesso, affronta il problema di definire alcune caratteristiche globali dell'editor suddividendole in tre ambiti logici.

Il menu Special – Le funzioni presenti in questo menu sono particolarmente interessanti. La prima voce serve a ingrandire, rimpicciolire e spezzare le finestre, e a muoversi da una finestra all'altra. L'opzione Format, invece, serve a eseguire correttamente il wordwrap sul bordo destro.

La voce DOS/ARexx interface serve a eseguire comandi dell'AmigaDOS o file comandi AREXX dall'interno di *CygnusEd*. A questa voce corrispondono cinque diverse opzioni. Le prime quattro, Send DOS/ARexx command, Install DOS/ARexx command, Load DOS/ARexx commands e Save DOS/ARexx commands, si spiegano da sole e soddisfano in pratica l'esigenza di gestire comandi dell'AmigaDOS e file comandi AREXX senza lasciare l'editor, sveltendo così le operazioni. La quinta, invece, serve per la definizione dei dieci tasti funzione dell'Amiga.

Con le prossime sei scelte di questo menu possiamo effettuare le seguenti operazioni: inserire nel file un carattere qualsiasi (anche non stampabile) indicandone il codice ASCII; spostare il testo all'interno della finestra; centrare una linea rispetto ai margini attivi della finestra; attivare ripetutamente la pressione di un tasto o la selezione di un menu; muovere il cursore alla parentesi che chiude quella su cui si trova.

Il menu Cut/Paste – Permette di selezionare blocchi di testo, e può eseguire le consuete operazioni a essi relative. Ce n'è una sola di atipica, Rot, che permette di crittografare un testo, rendendolo illeggibile se non si dispone della chiave per decifrarlo. Questa opzione è utile essenzialmente per la protezione dei file nelle trasmissioni all'interno di un network. Sempre nel menu Cut/Paste sono infine disponibili diverse opzioni per la cancellazione e il recupero di linee, parole e singoli caratteri.

Il menu Search/Replace – In aggiunta alle consuete operazioni di ricerca e sostituzione, consente l'inversione da maiuscolo a minuscolo (e viceversa)

dei singoli caratteri o d'interi parole.

Il menu Move – È stato ideato per permettere rapidi spostamenti all'interno della finestra attiva: abbiamo infatti la possibilità di spostare il cursore su un particolare punto del file precedentemente "etichettato" con l'opzione Auto-mark; ci si può anche spostare all'inizio o alla fine di una finestra, di una riga, dell'intero file; di 12 caratteri in qualsiasi direzione o alla parola successiva o precedente.

Conclusioni

CygnusEd Professional è l'editor che aspettavamo da sempre per l'Amiga: le sue caratteristiche ne fanno un ottimo pacchetto per la stesura di testi di qualsiasi tipo. È da sottolineare l'utilità del sistema a finestre multiple, che consente un colpo d'occhio immediato su più file o su diversi punti dello stesso file. Le finestre non sono comode come quelle di Intuition ma richiedono meno memoria, e del resto le finestre d'Intuition sarebbero forse troppo sofisticate per le esigenze di un editor.

Programmatori, giornalisti, scrittori e tutti quelli che scrivono professionalmente sanno bene quanto sia importante aumentare l'efficienza del proprio ambiente di lavoro. In *CygnusEd* troveranno un aiuto prezioso, oltre che completo e affidabile. Pur dovendo fare i conti con l'attuale risoluzione video disponibile sull'Amiga, *CygnusEd* offre innovazioni molto avanzate rispetto ai suoi diretti concorrenti: quando avremo a disposizione le nuove risoluzioni video promesse dall'ECS sarà difficile trovare rivali per questo prodotto. *CygnusEd*, infatti, è già pronto per sfruttare al meglio le future potenzialità dell'Amiga.

Il prossimo articolo

Nel prossimo articolo prenderemo in esame *Extend*, un programma che migliora l'Amiga BASIC, facendolo accedere alle potenzialità di Intuition, come i menu, i requester e i gadget. *Extend* aggiunge al Basic dell'Amiga circa 50 comandi allo scopo di gestire più facilmente le finestre, i menu, le combinazioni di tasti e i colori sullo schermo del *Workbench*. Inoltre permette di eseguire comandi dell'AmigaDOS dall'interno dei programmi in Basic, di utilizzare e gestire file grafici (anche in formato HAM) e fonti-carattere. Rende possibile usare dinamicamente la memoria, creare requester per i file e barre di scroll direttamente dai programmi in Basic. Esamineremo quindi tutte queste estensioni dell'Amiga BASIC e ne vedremo le applicazioni pratiche con esempi d'impiego reale. ■

Per ulteriori informazioni contattare direttamente:

ASDG Incorporated
(*CygnusEd Professional*, \$99)
925 Stewart Street
Madison, Wisconsin 53713, USA
(Tel. 001/608/2736585)

APPRETRATI

COMPLETATE LA VOSTRA COLLEZIONE

NUMERO 1/86 - Telecomunicazioni: guida all'acquisto di un modem. Il disk drive 1541 ed i suoi limiti tecnici. Corso di programmazione in L.M. su C-64. Il C-128: prova il nuovo Commodore Amiga a confronto con IBM PC, AT e Macintosh. Reset per il 1541. Amiga, dove fantasia e realtà si incontrano. Fare musica con il C-128. Convertire e grafico per Commodore 64. Ilistato. Cultura e informatica. Software Gallery: Lode Runner, Rescue, The Hitchhiker's Guide to the Galaxy, Jet, Scenery Disk, Software Helpline: The Hitchhiker's Guide to the Galaxy, Zaxxon, Wolfenstein, Star Wars, Ghostbusters, Jumpman, Mindshadow e Tracer Sancton.

NUMERO 2/86 - Jack Tramiel. Corso di programmazione in L.M. su C-64. Telecomunicazioni: il C-128D. Come operare all'interno del disk drive. Computer e grafica. La Commodore e la didattica. La Rom del C-64. Amiga e Atari 520ST a confronto. Speciale USA: il CES di Vegas, il modo del 1000. Il Commodore Show di San Francisco. La mappa di memoria del C-128. Fare musica con il 64. Rondò Veneziano. Archivio programmi: Ilistato per C-64. Software Gallery: Turbo Loading Cartridge, Machine Lighting Basic, Lightning, Urindum, Software Helpline: The Hitchhiker's Guide to the Galaxy, Beyond Castle Wolfenstein, Ghostbusters, Cosmic Balance, Jumpman, Star Trek, Avventura nel computer.

NUMERO 3/86 - Nuovi prodotti Commodore. C-64, 1581 e 1802. Linea Commodore PC IBM compatibilizzabile. Gli Ampersand file. Geos. Corso di programmazione in L.M. su C-64. Il SIM HI-FI IVES Software in CP/M per C-128. Amiga: una nuova era nel mondo del PC. La A-Squared e AmigaLive! Le piante parlano. progetto hardware/software. Amiga: tra sogno e realtà. Espansioni di memoria per il C-128. I disk drive 1570 e 1571 a confronto. Prom ed Ilistato. Il suono nel C-128. Jenny: 1541 Suoni e musica con l'AmigaBASIC. Software Gallery: Di-Sector V3.0, Project: Drawing Board, Textcraft, Game Killer, Nexus, Mach 128, 1571 Clone Machine, VizaStar e VizaWrite, The Final Cartridge. Software Helpline: The Dallas Quest, Quotron

NUMERO 4/86 - Smau '86. CLI Command Line Interface. La compatibilità IBM per Amiga. Sidelcar e Transformer. Schema audio e video con l'Amiga. Amiga days '86. Geos e il suo creatore. intervista: Personalizziamo il sistema operativo del C-64. Csa turbo Amiga. La memoria di massa e i compact disk. Terzo Commodore Show, Los Angeles. Tempo di Biennale, tempo di Amiga. Protezione scrittura e interruzione per il numero di device. progetto hardware per C-64 Software per Amiga. descrizione di più di 250 programmi. Grafica 128. Ilistato per C-128. Corso di programmazione in L.M. su C-64. Lista delle routine Kernel del C-128. Software Gallery Johnny Reb II. War Play, Time Trax, Aegis Impact, The Cataloger, Cartuccia ES-9, Onlined, Deluxe Paint. Software Helpline: Borrowed Time

NUMERO 1/87 - Il C-64 a Hollywood. L'Amiga nella realizzazione del serial "Amazing Stories". Amiga e Pontcaro. La politica economica della Commodore. Gli Idea Processor. World of Commodore servizio speciale. Amiga 1060 Sidelcar. La grafica. L'Amiga e l'AmigaBASIC. Grafica avanzata per il C-128. programmazione del chip 85633. Geos le chiavi del regno. Super! Basket NBA simulazione per C-64/128. Commodore PC 40. AT Hardcopy della pagina grafica 640x200 pixel per C-128. Gestione magazzino per C-128. Routine alternate Kernel-Disk. La stampante MPS 1000. Corso di programmazione in L.M. su C-64. Software Gallery Tomahawk Flight Simulator II. Part 128. Deluxe Paint II. Defender of the Crown. Blood n' guts. 43 one year after Mind Walker. Thai boxing. Skyfox. Software Helpline: Flight Simulator II. View to a kill. Ultima IV

NUMERO 2/87 - Il Consumer Electronics Show: servizio speciale. Nuovi nati nella famiglia Commodore: Amiga 500 e 2000. Nuove istruzioni del chip 8502 utilizzato come CPU nel Commodore 128. La videodigitizzazione: due noti digitalizzatori per C-64/128 e Amiga. Batch file con l'Amiga. Corso di programmazione in L.M. per C-64. La produzione di "The Bard's Tale": intervista in esclusiva. Gestione di un club: Ilistato per C-64. Protector 128: protettore di programmi per C-128. Geos: le chiavi del regno. Nuove potenzialità per le macchine a 8 bit della Commodore. Copie bit a bit sul 1541. Software Gallery: American Challenge e America's Cup, Page Setter, The Inheritance, Webstars, Vera Cruz, Agent Orange, S.D.I., Sinbad, Turbo Pascal, EDNA, Power Cartridge. Software Helpline: Co-Topos.

NUMERO 3/87 - Commodore Italiana: gli anni nella manica per il 1987. I nuovi Commodore Amiga 500 e 2000. CeBit '87: servizio speciale. Roma Ufficio '87. La famiglia di programmi Geos: le applicazioni dedicate e la nuova versione per C-128. Corso di Programmazione in L.M. su C-64. Find 128: utility per C-128. Commodore Show San Francisco: servizio speciale sulla nota rassegna californiana. Il Software Sistema dell'Amiga. Catalogazione Dischi per C-128: Ilistato per C-128. Il disk drive OC-118: prova hardware. Dump 128: Ilistato. Velocizzazioni e disk drive 1541. Scroll list 128: Ilistato. Software Gallery: Balance of Power, Silent Service, Portal, Graphic Adventure Creator, Meldy Hall's Printware Series, Cyborg, Twin Tornado, Thanatos, The Ket Trilogy. Software Helpline: Leather Goddesses of Phobos.



NUMERO 6/87 - SIM HI-FI. SMAU Dossier speciale: i pirati in Italia. La storia del gruppo 2703: un nuovo business. Il hardware pirata: un pirata pentito, la riscossa degli importatori, quando la copia costa più dell'originale. Il periodo dell'impunità è finito. Prove hardware: il disk drive da 3.5 1581 Telematica. I Videotel per il C-64. Amiga: le nuove frontiere dei giochi d'avventura. Toolkit 128: Ilistato per nuovi comandi BASIC. Corso di programmazione in L.M. per C-64 decima puntata. Il sistema BCD Amiga hardware gli sprite e il blitter. I nuovi prodotti USA per Amiga. Elab C-128: Ilistato di un programma grafico Software Gallery ACE 2. Pirates! Passengers on the wind. Lurking horror. Hollywood hi-jinx. Vampyr. Motel. I predatori della Valle del Re. Vizevite desktop. 007 The living daylight. Solomon's key. Software Helpline: The lurking horror. Bureaucracy



NUMERO 4/87 - Obiettivo Telematica: la telematica e il C-64. La Commodore cambia video: servizio speciale. Basic 8 bit: sistema grafico per Commodore 128. Desktop publishing con l'Amiga: rassegna di programmi di desktop publishing realizzati per l'Amiga. La stampante MPS 1200: prova hardware. Amiga: Ilistato in Amiga BASIC per tracciare il grafico di funzioni matematiche. Il Consumer Electronics Show di Chicago: Servizio speciale. Nuovi comandi e Ram Disk: Utility per C-128. Disk Sector Editor: Ilistato per C-128. L'hardware dell'Amiga. Bank Data: archivio di nomi e indirizzi per C-128. Corso di Programmazione in LM per C-64. Software Gallery: Bureaucracy, Faery Tale, Unwitted, Murder on the Atlantic, Deathscape, Frankenstein, Up Perciscope, Barbarian, 500 cc Grand Prix, Logistix, Superbase. Software Helpline: A View to a Kill.



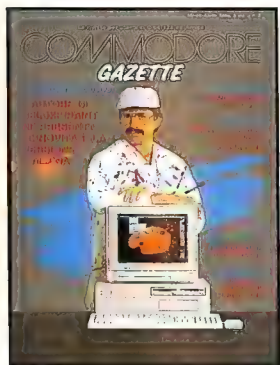
NUMERO 7/87 - Le promesse del 1988: rivista, a manager. della Commodore. Novità per Amiga dagli USA. AmieXPO, Commodore Show e Comdex. Anteprema sugli emulatori di C-64 per Amiga. L'Aids del computer. I programmi viri. Fred Fish: i programmi di pubblico dominio per Amiga. La geometria frattale. Le teorie di Mandelbrot. Il nuovo programma in Amiga Basic. Le espansioni di memoria 1764, 1700 e 1750 per C-64/128. Prova del PC1. Videocassette 3D: il desktop video C-128: l'avvenire si chiama Geos. Amiga hardware: ultima puntata. Corso di programmazione in LM per C-64: assemblatori e codice sorgente. Software Gallery Advanced OCP art studio. Alien hers. Echelon. Bubble bubble. Knight or. Quodex. Fire power. Test drive. Video title show. The hunt for Red October. Software Helpline: The three musketeers



NUMERO 5/87 - Computergrafica nella ricerca scientifica. Iuso dell'Amiga. Software musicale per l'Amiga. L'Amiga 2000 e la Compatibilità IBM: un'introduzione al modello 2000 e alla A2088. Inventando l'Amiga: il travagliato avvio commerciale del gioiello della Commodore. Corso di Programmazione in L.M. per C-64. L'Hardware dell'Amiga: le caratteristiche strutturali dell'Amiga. Il Comdex di Atlanta: servizio speciale. Codici in data C-128: Ilistato. Amiga 3D: animazione in 3D con l'Amiga. Append Merge: Ilistato per C-128. Semper Sperum: Ilistato per C-64 per la pianificazione delle giocate al Lotto. Bank Data. La produzione di Adventure nel nostro Paese. Software Gallery: The Mirror Hacker package, ProWrite, Guild of Thieves, Barbarian, Defender of the Crown, Denarius, Revs Plus, Inheritance 2, Kinetik, The Three Musketeers, Galileo. Software Helpline: Trinity



NUMERO 1/88 - Novità dagli Usa per l'Amiga. AmieXPO e World of Commodore Show. Intervista con Dale Luck sui nuovi Kickstart e Workbench 1.3. Speciale telematica: il 6499 alla prova. Amiga e Videotel: intervista con Hugo Cornwall. La telematica in 60 parole chiave. W e info file a cosa servono e come usufruirne. Un viaggio nell'universo dei giochi di ruolo: il quinto convegno Automail. Geos: anatomia di un sistema operativo. Corso di programmazione in LM per C-64. Equo canone C-64: Ilistato. Gli astri attraverso il C-128: Ilistato. Software Gallery Airbone ranger. Apollo 18 mission to the moon. Chuck Yeager's advanced flight trainer. Auto drive, Moebius, Shoot out, up construction. The art of chess. World tour golf. Bocce, Tombola, Flying shark, Miniputt, Crazy car. Fest drive. Software Helpline: Neverending story. Stationfall



NUMERO 2/88 - Intervista con Dan Silva, l'autore di Deluxe Paint. Speciale simulazione del volo: Flight Simulator II, il funzionamento di Flight Simulator II, Co-Pilot un libro da volare, il volo a schema, simulatori di volo a confronto. Reportage dal CEBIT 88, la più grande fiera europea dell'informatica. Sconfitta l'AIDS dell'Amiga, con il listino del programma antivirus "Guardian". Prova hardware: hard disk per l'Amiga, geoPublish, il desktop publisher GEOS compatible. Corso di programmazione in LM per l'C-64, Panacea Basic C-128, listato, SUPER Quark C-128, listato, il progetto ARP Software Gallery: Shadowgate, Stratton, Yogi Bear, Terramex, Ports of Call, The Graphics Studio, Jet, John Brenner Boston Celtic, Power at Sea, Stealth Fighter, Card Sharks, Mercenary, Strike Fleet, King of Chicago, Jinxter Software Helpline: Perry Mason and the Case of the Mandarin Murderer.



NUMERO 3/88 - I programmatori del mondo Amiga a confronto. I PC Commodore IBM compatibili della terza serie. Una rete telematica sul mondo. Come crearsi un videogioco "fatto in casa". Come far pubblicare un programma. La programmazione in ambiente Geos con Geoprogammer. Suddividiamo in moduli i nostri programmi Basic. Geos: anatomia di un sistema operativo. Un archivio elettronico per il vostro Amiga. I drive ottici: trenta hard disk da 20 MB in un CD Rom. Corso di programmazione in LM per C-64. Software Gallery: Shakespeare, Video Titler, Corporation, Ferrari Formula One, Geos 117, Return to Atlantis, Rimrunner, Intelligence, Skyfox II: The Cygnus Conflict, Crack, Io, Advanced Tactical Fighter, Tetris, Troll, Software Helpline: John Brenner, Boston Celtic, Beyond Zork, Curiosità e informazioni utili.



NUMERO 4/88 - Videoscope 2.0 ■ le nuove frontiere dell'immagine. L'Amiga, le immagini, la Polaroid Palette. Gestire i testi in ambiente Geos: Geowrite Workshop. La MPS 1500 C: i colori economici. Il servizio Videotex nel mondo. Il più grande software del mondo. CES: I nuovi videogiochi alla moda di Chicago. Geos: anatomia dei menu ■ delle icone. Un C-128, un telescopio e una stella cometa. Corso di programmazione in L.M. per il C-64. A caccia di immagini nella memoria del C-64. Software Gallery: Laserterm, Vixen, Express Paint, The Three Stooges, Patton vs Rommel, BeckerBasic, Personal Newsletter, Enlightenment Druid II, Publisher Plus, Sentinel, F/A 18 Interceptor, Capone, The Armageddon Man, Question II, Software Helpline: Labyrinth, Shadowgate, Curiosità e informazioni utili.



NUMERO 5/88 - ...E la Sublogic creò il volo. Matematica a tre dimensioni per l'Amiga. Okimate 20, una stampante dai colori brillanti. Un database per il mondo di Geos. Un tavolo da disegno per l'Amiga. La gestione multitasking, i tipi di codice e gli interrupt. Chicago: l'hardware e il software all'Amiexpo '88. Geos: anatomia di un sistema multitasking. Corso di programmazione in LM per il C-64. Software Gallery: Bard's Tale II, Carner Command, Breach, Halls of Montezuma, Wasteland, PKMate, Legend of the Sword, Whirlgig, Corruption, The Presence is Missing Software Helpline: Che cos'è un'avventura? Curiosità e informazioni utili.



NUMERO 6/88 Schede acceleratrici per un Amiga "turbo". Le stampanti Manne-smann MT 87 ■ MT 88. Novità Commodore non certo troppe. Una visita al mondo Commodore. Le stampanti Fujitsu DX2300 e DL3300. Un foglio elettronico per il mondo di Geos. Grafica avanzata con le espansioni RAM 1700 e 1750. Corso di programmazione in C per l'Amiga. Come programmare i gadget stringa di Intuition. La struttura ExecBase dell'Amiga Software Gallery Ultima IV, Ultima I, Starfleet I, Bobo, Game Over II, Caveman, Ughlympics, Bubble Ghost, Barbarian II, Starfighter 2, Red Storm Rising, Fusion, Virus, Rocket Ranger, Typhoon Software Helpline: Sherlock Holmes.



NUMERO 1/89 - L'intelligenza Artificiale sull'Amiga ■ il pacchetto Magellan. Le stampanti OKI a 9 aghi e a 18 aghi. La parola ai programmatori sta nascendo l'Amiga 3000? Geos come costruire un box di dialogo. Un C-128 a caccia di eclissi. Corso di programmazione in C per l'Amiga. Come programmare i gadget booleani di Intuition. ExecBase: i parametri dinamici e le liste di sistema. Le macro istruzioni Assembly dell'Amiga Software Gallery, Battle Chess, Microprose Soccer, The Universal Military Simulator, Who framed Roger Rabbit, Neuro-mancer, Deathlord, Driller, Falcon, Times of Lore, Captain Blood, Elite, The Mars Saga, Total Eclipse, Simulgolf Software Helpline: Ultima V, Pool of Radiance.

✂

NOME

COGNOME

INDIRIZZO

CITTÀ

C.A.P.

DESIDERO RICEVERE I SEGUENTI NUMERI ARRETRATI:

PER UN TOTALE DI LIRE (lit. 16.000 ognuno):

ALLEGO ASSEGNO BANCARIO D CIRCOLARE

ALLEGO FOTOCOPIA DELLA RICEVUTA DEL VAGLIA POSTALE

FIRMA.....

RITAGLIARE E SPEDIRE A: COMMODORE GAZETTE
Via Monte Napoleone, 9 - 20121 Milano

I GADGET PROPORZIONALI DI INTUITION

I gadget proporzionali sono fra i più interessanti di Intuition. L'articolo e il programma dimostrativo spiegano come funzionano e come devono essere gestiti dai programmi

di Harriet Maybeck Tolly

Eccoci al terzo e ultimo articolo dedicato all'analisi dei gadget di Intuition. Dopo aver esaminato i gadget stringa nel numero 6/88 di *Commodore Gazette* e i gadget booleani nel numero scorso, siamo arrivati finalmente ai gadget proporzionali.

In genere questo tipo di gadget viene utilizzato per mettere a disposizione dell'utente alcune barre sulle quali può agire per fare scorrere una serie di dati sullo schermo, ma servono anche per fornire informazioni in un modo particolarmente intuitivo. Esempi di gadget proporzionali che tutti conoscono sono le barre di scroll utilizzate dal *Workbench* per le sue finestre, che fanno scorrere il contenuto della finestra e inoltre indicano le proporzioni dell'area visibile rispetto all'area non visibile. Un altro esempio di gadget proporzionale potrebbe essere una barra disposta verticalmente lungo la finestra di un editor: la lunghezza della barra mostra quale percentuale di testo è visibile nella finestra mentre la sua posizione indica qual è la posizione del testo visibile rispetto all'intero file. Spostando il gadget con il mouse si potrebbe poi effettuare lo scroll del testo.

I cursori dei gadget proporzionali possono spostarsi sia in verticale sia in orizzontale, anche contemporaneamente. Oltre a essere assai versatili, e quindi particolarmente utili, questi gadget sono anche quelli che nell'ambito di Intuition sembrano avere meno bug. Purtroppo, però, molti programmatori si trovano spesso in difficoltà nel comprendere come vanno utilizzati certi loro parametri che riguardano la forma, le dimensioni, la massima altezza di spostamento, la massima larghezza di spostamento e così via.

Enteremo quindi, prima di tutto, di descrivere questi parametri dettagliatamente.

La struttura PropInfo

Come tutti i gadget, anche quelli proporzionali devono essere definiti tramite una struttura *Gadget*, all'interno della quale occorre però memorizzare (nel parametro *SpecialInfo*) l'indirizzo di una struttura ausiliaria specificamente richiesta dai gadget proporzionali, *PropInfo*. Eccone la definizione:

```
struct PropInfo
{
    USHORT Flags;
    USHORT HorizPot;
    USHORT VertPot;
    USHORT HorizBody;
    USHORT VertBody;
    USHORT CWidth;
    USHORT CHeight;
    USHORT HPotRes, VPotRes;
    USHORT LeftBorder;
    USHORT TopBorder;
};
```

Analizziamo uno per uno i parametri della struttura. Lo stato di quattro bit del parametro **Flags** viene impostato dal task, mentre un quinto bit viene impostato da Intuition. Il task può impostare **AUTOKNOB** (rende variabile l'ampiezza della barra e le assegna la forma standard gestita da Intuition); **FREEHORIZ** (permette il movimento orizzontale); **FREEVERT** (permette il movimento verticale); **PROPBORDERLESS** (elimina la cornice). Il flag **KNOBHIT** viene invece impostato da Intuition quando l'utente seleziona il gadget con il mouse: serve al task per rilevare se il movimento del gadget viene provocato "afferrandolo" con il mouse o premendo il tasto del mouse nell'area del gadget non occupata dalla barra. Sui flag **FREEHORIZ/VERT** occorre aprire una parentesi: talvolta può presentarsi l'esigenza d'impiegare i gadget

proporzionali come semplici strumenti di output per le applicazioni, cioè come indicatori, senza che l'utente possa agire sulle barre. È opinione diffusa (ma errata) che per ottenere questo funzionamento non si debba impostare né il flag FREEHORIZ né il flag FREEVERT. Bisogna invece disabilitare il gadget impostando il flag GADGDISABLED nel parametro Flags della sua struttura Gadget.

I parametri **HorizPot** e **VertPot** possono essere impostati dal task per indicare la posizione iniziale della barra all'interno del gadget; se la barra è libera di muoversi nel range almeno una direzione, Intuition provvede poi a mantenere aggiornato il relativo parametro a seconda di come viene spostata. Il valore dev'essere compreso fra 0x0000 e 0xFFFF: per esempio, nel caso di HorizPot, il valore minimo colloca la barra tutta a sinistra all'interno del gadget, mentre il valore massimo la colloca tutta a destra; i task di solito calcolano il valore opportuno nel range citato per riflettere con la posizione della barra la situazione iniziale. Nel caso dell'editor che avevamo portato come esempio, dovrebbe essere mantenuto aggiornato il valore contenuto nel parametro VertPot.

I parametri **HorizBody** e **VertBody** si riferiscono invece alle dimensioni della barra in relazione alla quantità d'informazioni (o di finestra) visibile sullo schermo. Anche per loro i limiti sono 0x0000 e 0xFFFF. Sempre nel caso dell'editor che abbiamo citato come esempio, il valore contenuto in VertBody dovrebbe essere mantenuto aggiornato per riflettere la dimensione della parte di testo visibile nella finestra rispetto alla dimensione dell'intero testo.

Tutti i parametri successivi, vengono aggiornati da Intuition: **CWidth** e **CHeight** rappresentano la larghezza e l'altezza del contenitore all'interno del quale può muoversi la barra. **HPotRes** e **VPotRes** contengono un valore pari al doppio dell'incremento orizzontale o verticale che la barra subisce in maniera automatica quando l'utente la muove senza afferrarla (generalmente questo incremento è pari alla lunghezza della barra). **LeftBorder** e **TopBorder** indicano rispettivamente lo spessore del bordo sinistro e del bordo superiore del gadget.

Il funzionamento dei gadget proporzionali

Un gadget proporzionale è composto di due parti: quella mobile è la barra (in inglese "knob", manopola), mentre l'area dove si muove la barra viene chiamata "contenitore". Mascherandone alcune parti, i task possono fare in modo che il contenitore assuma forme particolari, per quanto il computer lo "veda" sempre come un rettangolo. Per quanto riguarda la forma della barra, i task possono indicarne una personalizzata; se invece non si desiderano forme particolari, è sufficiente impostare il flag AUTOKNOB del parametro Flags: basterà dichiarare una struttura di tipo Image (senza inicializzarla) perché sia

Intuition a definire la forma del gadget.

Il programma in C che accompagna l'articolo mostra alcuni possibili impieghi dei gadget proporzionali. Il più comune è quello del gadget che offre all'utente una barra per scorrere un elenco di stringhe.

I parametri Body e Pot

I task devono sempre impostare il valore di **Body**, che rappresenta le dimensioni della barra. Se deve muoversi verticalmente, occorre impostare il parametro VertBody e il flag FREEVERT; se deve muoversi orizzontalmente occorre impostare il parametro HorizBody e il flag FREEHORIZ. A questo proposito è opportuno notare che sebbene i flag FREEVERT e FREEHORIZ permettano di limitare il movimento della barra in una delle due direzioni, i valori del puntatore restituiti nelle variabili IntuiMessage.MouseX e IntuiMessage.MouseY continueranno a variare entrambi in ogni caso. Per esempio, varierà anche la coordinata orizzontale anche se è impostato soltanto il flag FREEVERT e la barra si muove solo verticalmente.

In genere i valori da assegnare ai parametri Body devono rappresentare la percentuale di dati visibili rispetto alla quantità totale. Nel nostro programma, la quantità di stringhe visibili è 4 e il totale è 10. I valori che i parametri Body possono assumere vanno da 0x0001 a 0xFFFF, rispettivamente la più piccola e la più grande barra visualizzabile. Se tutti i dati sono visibili il valore sarà 0xFFFF, se ne è visibile solo la metà occorrerà impostare Body a 0x7FFF, e così via. La formula che generalmente si adotta è di questo tipo:

$$\text{Body} = (\text{ULONG})(\text{STRINGHE-VISIBILI} * 0xFFFF) / \text{STRINGHE-TOTALI};$$

Il valore di Body determina anche di quanto si muove la barra quando l'utente preme il pulsante del mouse al suo esterno. Per esempio, se Body vale 0x7FFF, la barra riempie la metà del contenitore: in un gadget a movimento verticale, premere il pulsante del mouse sotto la barra produrrà il suo spostamento nella metà vuota, ossia il movimento avviene in passi del 50 per cento. Questo va bene se abbiamo un gadget standard, con il flag AUTOKNOB impostato, ma potrebbe non essere desiderabile se abbiamo definito un gadget personalizzato che occupa solo il 10 per cento del contenitore; occorre quindi ritoccare opportunamente il valore contenuto in Body.

I parametri **Pot** (da "potenziometro") possono assumere lo stesso insieme di valori dei precedenti: quando la barra è tutta in alto o a sinistra, il valore di Pot è zero, mentre se si trova nella posizione opposta il suo valore è 0xFFFF. Intuition non fornisce nessun sistema per invertire il valore di questo parametro, sebbene sia facile applicare la formula: $\text{Valore-Invertito} = 0xFFFF - \text{Pot}$

In genere, Pot viene inizializzato a 0x0000, tuttavia in alcuni casi può esserci la necessità d'impostare diversamente il suo valore iniziale.

cioè fare in modo che la barra inizialmente non si trovi al limite superiore, o sinistro, del contenitore. Se si sa qual è la prima stringa che si vuole visualizzare, e il suo numero è stato memorizzato nella variabile `STRINGHE_INIZIALE`, si può applicare la seguente formula per ricavare il valore di Pot: `Pot = MIN (0xFFFF, ((ULONG)STRINGHE_INIZIALE << 16)/(STRINGHE_TOTALI - STRINGHE_VISIBILI))`;

In questa formula, per sicurezza, viene impiegata la macro istruzione `MIN` (questa macro è definita nel file `INCLUDE clib/macros.h`).

Infine, quando il task riceve un messaggio generato dal fatto che l'utente ha spostato la barra all'interno del contenitore, i valori dei parametri Pot variano. Nel nostro programma assumiamo che un valore di Pot pari a `0x0000` corrisponda al primo dei dieci nomi di cui disponiamo: un valore pari a `0xFFFF` corrisponde quindi all'ultimo. Quando l'utente sposta la barra, provocando la modifica dei parametri Pot, il programma determina il numero della nuova stringa da visualizzare attraverso la seguente formula: `STRINGHE_INIZIALE = (Pot * (ULONG)(STRINGHE_TOTALI - STRINGHE_VISIBILI) + (1 << 15)) >> 16`. Esistono altre formule per utilizzare il valore Pot, ma sono tutte di questo tipo, sebbene alcune eseguano più controlli su eventuali overflow e altre siano più veloci. Quella proposta rappresenta un'opportuna via di mezzo.

Lo spostamento della barra

Vediamo come si fa a rilevare che l'utente ha mosso la barra dalla sua posizione. Un modo molto semplice consiste nell'impostare il flag `RELVERIFY` del gadget interessato, il che costringe Intuition a inviare al programma un messaggio `IDCMP` di tipo `GADGETUP` quando il pulsante sinistro del mouse viene rilasciato. A questo punto è possibile leggere il nuovo valore di Pot e modificare di conseguenza l'elenco delle stringhe.

Sebbene il manuale di Intuition affermi che impostando il flag `RELVERIFY` viene generato un messaggio `GADGETUP` solo se il puntatore si trova ancora sul gadget, questo NON è vero nel caso dei gadget proporzionali. Una volta che il pulsante sinistro viene premuto sopra il gadget, allontanando il puntatore dalla barra con il pulsante premuto, la barra continua comunque a seguirne i movimenti; quando infine il pulsante viene rilasciato, anche se il puntatore non è più sulla barra, si otterrà ugualmente un messaggio di tipo `GADGETUP`.

Un altro metodo consiste nell'aggiornare le stringhe sullo schermo durante gli spostamenti della barra, senza attendere che il pulsante venga rilasciato. Sebbene sia più complicato, è il metodo che abbiamo seguito nel programma che accompagna l'articolo. Occorre impostare il flag `FOLLOWMOUSE` del gadget, oltre a `GADGIMMEDIATE` e `RELVERIFY`, di modo che al programma giungano anche i messaggi `MOUSEMOVE` di

Intuition. Con questa configurazione, i messaggi provenienti da Intuition arriveranno nell'ordine seguente: prima `GADGETDOWN`, non appena l'utente preme il pulsante sinistro del mouse con il puntatore sulla barra, poi una serie di messaggi `MOUSEMOVE` a mano a mano che la barra viene spostata; ogni volta che non risultano più messaggi accodati il ciclo di elaborazione termina, e il ciclo principale del programma (il ciclo `FOREVER`) prosegue a leggere il valore del parametro Pot e ad aggiornare opportunamente la lista delle stringhe sullo schermo. Si noti che l'aggiornamento della lista avviene senza che sia pervenuto il messaggio `GADGETUP`.

C'è un particolare che vorremmo mettere in evidenza. Se il flag `GADGIMMEDIATE` non è impostato, quando l'utente seleziona la barra del gadget proporzionale il programma riceve il messaggio `MOUSEMOVE` anziché il consueto `GADGETDOWN`. Questa è una situazione anomala, in quanto non è detto che la selezione della barra ne causi anche il movimento, come invece viene indicato dal messaggio `MOUSEMOVE`.

Particolarità e stranezze dei gadget proporzionali

Un piccolo quiz: chi sa che cosa sono i fuzzy? Per coloro che non hanno seguito i precedenti articoli di questa serie, diciamo che si tratta di pixel che appaiono dove non dovrebbero in seguito alla trasformazione di un gadget in fantasma. Abbiamo già visto che con i gadget stringa ottenevamo un po' di fuzzy quando la larghezza del gadget non era esattamente un multiplo dalla larghezza dei caratteri utilizzati; i gadget proporzionali hanno invece un loro tipo di fuzzy tutto particolare e ancora più diabolico.

Per vedere il bug in azione, si devono impostare i colori del testo e dello sfondo in maniera tale che non corrispondano ai colori di default dello schermo su cui si trova il gadget. In altre parole, sullo schermo del *Workbench* non va usata la combinazione (0,1), e nemmeno (-1,-1) che provocherebbe ugualmente l'uso dei valori di default. Così facendo, se nella lista dei gadget della finestra aperta dal programma esiste un gadget proporzionale che punta a un gadget stringa disabilitato, il suo contenitore verrà riempito di fuzzy. Funzionerà benone, ma sarà mascherato come avviene con un gadget fantasma. Le soluzioni a questo problema sono l'uso dei colori di default, oppure l'assoluta sicurezza che nessun gadget proporzionale punti a un gadget stringa disabilitato (fantasma). Nel programma d'esempio che pubblichiamo mostriamo agli increduli gli effetti di questa disavventura grafica.

Gadget sul bordo

Sistemare alcuni gadget sul bordo di una finestra è un'idea attraente, ma può risultare di

difficile realizzazione. Il manuale di Intuition dice che «i gadget si possono tranquillamente disporre lungo i bordi di una finestra» e un utente ha tutto il diritto di pensare che i testi vengano automaticamente troncati in corrispondenza del bordo interno del gadget, come accade con i bordi delle normali finestre. Purtroppo, non è così che vanno le cose. Se infatti scriviamo un testo (IntuiText) nella finestra, questo non viene affatto troncato in corrispondenza di un gadget di bordo. Vediamo come si può ovviare all'inconveniente.

Indipendentemente dal fatto che la finestra in uso sia una finestra di tipo GimmeZeroZero, i valori contenuti nei parametri GZZWidth e GZ-ZHeight della struttura Window corrispondono alle reali dimensioni interne della finestra, tenendo conto dei gadget sul bordo: questo vuol dire che il taglio dei testi può essere effettuato direttamente in base a questi valori anziché facendo la sottrazione tra le dimensioni della finestra e la larghezza dei gadget. Supponendo di avere un gadget proporzionale sul bordo destro della finestra, come accade di solito, potremo formattare un testo a sinistra direttamente in base al valore riportato da GZZWidth.

Le estensioni della versione 1.2

Dal momento che pubblicazioni tecniche sulla versione 1.3 del software sistema non sono ancora disponibili, e la documentazione attuale risale alla versione 1.1, andiamo a rinfrescare la memoria sul alcune migliorie introdotte dalla versione 1.2, che peraltro non sono molto note alla grande utenza.

Nella versione 1.2 del software sistema è stata inserita una nuova versione della funzione ModifyProp(), chiamata NewModifyProp(), che è piuttosto interessante. Infatti la precedente versione richiamava durante l'esecuzione la vecchia funzione RefreshGadgets() la quale, come si è detto negli articoli precedenti, causava il flickering dei gadget poiché effettuava sempre e in ogni caso la rivisualizzazione di tutta la lista dei gadget, anche quando non era necessario. Con la nuova routine è possibile specificare il numero di gadget da sottoporre al refresh.

Si può inoltre specificare il flag GADGHIMAGE per informare Intuition che vogliamo usare un'immagine alternativa per lo stato di evidenziato della barra. Il manuale consiglia di utilizzare un'immagine delle stesse dimensioni della barra non selezionata: chi scrive ha provato sia a impostare un'immagine più grande sia a impostarne una più piccola, e in entrambi i casi l'immagine evidenziata non è risultata corretta.

Il programma

Il programma in C che accompagna l'articolo apre una finestra con quattro gadget proporzionali. Il primo serve a scorrere un elenco di dieci stringhe all'interno di uno spazio che ne può

contenere solo quattro; ovviamente, il programma tiene conto del numero di stringhe visualizzabili per dimensionare l'ampiezza della barra proporzionalmente a quella massima. Inoltre, si noti che il programma inizia visualizzando per prima la stringa numero 4 e collocando la barra di conseguenza; alterando il valore della costante STRINGA-INIZIALE nel sorgente e ricompilando il programma si può osservare come cambia la posizione della barra (si noti che questa costante può variare soltanto fra i limiti indicati nel sorgente); alterando il valore della costante STRINGHE-VISIBILI nei limiti indicati si può invece osservare come cambia l'altezza della barra.

Il secondo gadget mette in luce il problema dei fuzzy che si verifica quando il gadget proporzionale è legato a un gadget stringa disabilitato.

Il terzo dispone di una barra personalizzata con un'immagine alternativa.

L'ultimo gadget contiene una barra personalizzata libera di muoversi sia verticalmente sia orizzontalmente; è un gadget proporzionale che è stato definito con il contenitore privo di bordo. Nella sua gestione, il programma mostra all'utente i valori dei parametri Pot.

L'ultimo particolare da sottolineare prima di passare al programma vero e proprio, riguarda l'apertura di uno schermo personalizzato. Ai fini del funzionamento dei gadget questo nuovo schermo è del tutto inutile; è stato inserito solo per disporre di un colore in più rispetto ai quattro colori disponibili sullo schermo del *Workbench*, e per dare la possibilità all'utente di verificare che aumentando il numero di bit plane di questo schermo, e quindi il numero di colori disponibili, gli aggiornamenti dello schermo rallentano considerevolmente quando si muove un elemento grafico (come per esempio la barra mobile dell'ultimo gadget). Se nella struttura NuovoSchermoDiControllo si variasse il quinto parametro da 3 a 4 e si ricompilasse il programma, si noterebbe un ulteriore rallentamento nella gestione grafica dei gadget.

Listato: Gadget proporzionali

Programma che mostra la gestione dei gadget proporzionali.

E' stato compilato con il compilatore Aztec C della Manx, e necessita almeno della versione 1.2 per funzionare correttamente; deve essere mandato in esecuzione da CLI.

N.B.: tutti i dati delle immagini devono risiedere nella memoria chip (i primi 512K). Se si usa il compilatore Lattice, prima della fase di link occorre sottoporre il codice oggetto all'utility ATOM; se invece si usa il Manx bisogna selezionare l'opzione +cdb del linker.

Il programma e' stato realizzato da Harriet Maybeck Tolly; e' di pubblico dominio e come tale puo' essere distribuito gratuitamente.

```

#include <exec/types.h>
#include <exec/exec.h>
#include <intuition/intuition.h>
#include <graphics/gfxbase.h>
#include <functions.h>
#include <lib/macros.h>

/* Le tre costanti che seguono si riferiscono al gadget
 * proporzionale che consente di scorrere #STRINGHE_TOTALI
 * stringhe visualizzandone #STRINGHE_VISIBILI per volta.
 * Variandole nei range consentiti si puo' osservare il
 * funzionamento del gadget */

/* Numero di stringhe visibili
 * Range: 1L -- (STRINGHE_TOTALI - 1) */
#define STRINGHE_VISIBILI 4L

/* Numero di stringhe da visualizzare
 * Range: 1L -- 10L */
#define STRINGHE_TOTALI 10L

/* Prima stringa da visualizzare
 * Range: 0 -- (STRINGHE_TOTALI - STRINGHE_VISIBILI) */
#define STRINGA_INIZIALE 4

#define DEBUG TRUE /* Eliminare questa riga se non si */
                  /* desidera il controllo sulle */
                  /* costanti appena definite */

#define LUNGHEZZA_MAX 20 /* Lunghezza max stringhe */
#define PROP_LIBERO 1 /* ID gadget a movimento libero */
#define PROP_SCROLL 2 /* ID gadget di scroll stringhe */

struct IntuitionBase *IntuitionBase = NULL;
struct GfxBase *GfxBase = NULL;
struct Window *FinestraDiControllo = NULL;
struct Screen *SchemaDiControllo = NULL;
struct IntuiMessage *MioMessaggio;

/*.....
 * Dichiaro le strutture per i gadget proporzionali */
/*.....

/* Questo gadget mostra come interpretare i parametri Pot
 * per effettuare lo scroll di una serie di stringhe */

struct Image g_image_scroll;
struct PropInfo g_prop_scroll;

struct Gadget PropGadget_scroll = {
    NULL, /* Puntatore al gadget successivo */
    180, 30, 20, 60, /* Posizione e dimensioni del contenitore */
    GADGCHOMP, /* Flag di controllo */
    GADGIMMEDIATE|FOLLOWMOUSE|RELVERIFY, /* e di attivazione */
    PROPGADGET, /* Tipo di gadget */
    (APTR)&g_image_scroll, /* Puntatore alla struttura Image */
    NULL, /* Nessun ptr a SelectRender */
    NULL, /* e a GadgetText */
    0L, /* Niente MutualExclude */
    (APTR)&g_prop_scroll, /* Puntatore a SpecialInfo */
    PROP_SCROLL, /* ID del gadget */
    NULL /* Nessun ptr a dati speciali */
};

/* Questo gadget proporzionale punta a un gadget stringa
 * disattivato per mostrare il problema del fuzzy */

struct StringInfo StrInfo = {
    (UBYTE *)"Stringa immobile", /* Puntatore al buffer */
    NULL, /* Puntatore al buffer undo */
    0, /* Posizione del cursore nel buffer */
    17, /* Numero massimo di caratteri (NULL incluso) */
    0, /* Primo carattere da visualizzare */
    0, /* Posizione del carattere nel buffer undo */
    0, /* Attuale numero di caratteri nel buffer */
    0, /* Numero massimo di caratteri visibili */
    0, 0, /* Coordinate relative del contenitore */
    NULL, /* Nessun puntatore alla struttura Layer */
    0, /* Non e' un gadget stringa di tipo LongInt */
    NULL /* Nessuna mappa di tastiera alternativa */
};

struct Gadget StrGadget = {
    &PropGadget_scroll,
    220, 30, 160, 7,
    GADGCHOMP | GADGDISABLED,
    GADGIMMEDIATE | RELVERIFY,
    STRGADGET,
    NULL, NULL, NULL, 0, (APTR)&StrInfo, 0, NULL
};

struct Image g_image_fuzzy;

```

```

struct PropInfo g_prop_fuzzy;

struct Gadget PropGadget_fuzzy = {
    &StrGadget,
    400, 30, 20, 60,
    GADGCHOMP,
    GADGIMMEDIATE | RELVERIFY,
    PROPGADGET,
    (APTR)&g_image_fuzzy,
    NULL, NULL, 0, (APTR)&g_prop_fuzzy, 0, NULL
};

/*.....
 * Tutte le immagini devono risiedere nella memoria
 * chip (512K inferiori). Se il compilatore e il linker
 * non dispongono di un'opzione per questo scopo,
 * occorre allocare dinamicamente della memoria chip e
 * copiarvi i dati prima di assegnare i valori dei
 * puntatori ai parametri della struttura gadget
 *.....

UWORD PrimaImmagine[] =
{
    /* Bit plane 1 */
    0x0200, 0x0700, 0x0700, 0x0F00, 0x0F00, 0x1FC0,
    0x1FC0, 0x3FE0, 0x7FF0, 0x3FE0, 0x1FC0, 0x1FC0,
    0x0F00, 0x0F00, 0x0700, 0x0700, 0x0200,

    /* Bit plane 2 */
    0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
    0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
    0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF
};

UWORD ImmagineAlternativa[] =
{
    /* Bit plane 1 */
    0x7038, 0x3870, 0x1CE0, 0x0FC0, 0x0FC0, 0x0FC0,
    0x0FC0, 0x0FC0, 0x0FC0, 0x0FC0, 0x0FC0, 0x0FC0,
    0x0FC0, 0x0FC0, 0x1CE0, 0x3870, 0x7038,

    /* Bit plane 2 */
    0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
    0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
    0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF
};

struct Image g_image_primary = {
    1, 0, /* Coordinate vertice in alto a sinistra */
    13, 17, 2, /* Altezza, larghezza, # bit plane */
    PrimaImmagine, /* Puntatore ai dati grafici */
    0x03, /* PlanePick = piani di bit 1 e 2 */
    0x00, /* PlaneOnOff = nessun piano di bit */
    NULL /* Non seguono altre immagini */
};

struct Image g_image_alternate = {
    1, 0, 13, 17, 2,
    ImmagineAlternativa,
    0x03, 0x00, NULL
};

struct PropInfo g_prop_image;

struct Gadget PropGadget_image = {
    &PropGadget_fuzzy,
    500, 30, 21, 60,
    GADGIMAGE | GADGIMAGE,
    GADGIMMEDIATE | RELVERIFY,
    PROPGADGET,
    (APTR)&g_image_primary,
    (APTR)&g_image_alternate,
    NULL, 0, (APTR)&g_prop_image, 0, NULL
};

struct Image g_image_free = {
    1, 0, 13, 17, 2,
    ImmagineAlternativa,
    0x03, 0x00, NULL
};

struct PropInfo g_prop_free;

struct Gadget PropGadget_free = {
    &PropGadget_image,
    200, 120, 80, 35,
    GADGNONE | GADGIMAGE,
    GADGIMMEDIATE | RELVERIFY,
    PROPGADGET,
    (APTR)&g_image_free, NULL, NULL, 0,
    (APTR)&g_prop_free, PROP_LIBERO, NULL
};

```

```

);
.....
* Array di strutture IntuiText per la lista di stringhe *
* che vengono sottoposte allo scroll quando viene mossa *
* la barra del gadget. L'array e' dimensionato per un *
* massimo di STRINGHE_TOTALI stringhe visualizzabili *
.....

struct IntuiText IntuiStr[STRINGHE_TOTALI] = {
    {2, 3, JAM2, 5, 30, NULL, NULL, NULL},
    {2, 3, JAM2, 5, 40, NULL, NULL, &IntuiStr[0]},
    {2, 3, JAM2, 5, 50, NULL, NULL, &IntuiStr[1]},
    {2, 3, JAM2, 5, 60, NULL, NULL, &IntuiStr[2]},
    {2, 3, JAM2, 5, 70, NULL, NULL, &IntuiStr[3]},
    {2, 3, JAM2, 5, 80, NULL, NULL, &IntuiStr[4]},
    {2, 3, JAM2, 5, 90, NULL, NULL, &IntuiStr[5]},
    {2, 3, JAM2, 5, 100, NULL, NULL, &IntuiStr[6]},
    {2, 3, JAM2, 5, 110, NULL, NULL, &IntuiStr[7]},
    {2, 3, JAM2, 5, 120, NULL, NULL, &IntuiStr[8]}
};

.....
* Stringhe di testo che vengono fatte scorrere. *
* Sebbene in un'applicazione reale esse potrebbero *
* cambiare durante l'esecuzione, nel nostro esempio *
* questo non accade. Tuttavia non le completiamo con *
* gli spazi in fase di dichiarazione per mostrare *
* come compiere questa operazione in run-time *
.....

char stringhe_di_testo [STRINGHE_TOTALI][LUNGHEZZA_MAX] = {
    {"Prima stringa"},
    {"Seconda stringa"},
    {"Terza stringa"},
    {"Quarta stringa"},
    {"Quinta stringa"},
    {"Sesta stringa"},
    {"Settima stringa"},
    {"Ottava stringa"},
    {"Nona stringa"},
    {"Decima stringa"}
};

.....
* Testi che illustrano i gadget sulla finestra *
.....

struct IntuiText nomi[6] = {
    {2, 0, JAM2, 5, 15, NULL,
     (UBYTE *)"Scroll di stringhe", NULL},
    {2, 0, JAM2, 220, 15, NULL,
     (UBYTE *)"Gli indesiderati fuzzy!", &nomi[0]},
    {2, 0, JAM2, 430, 15, NULL,
     (UBYTE *)"Immagine evidenziata", &nomi[1]},
    {2, 0, JAM2, 170, 108, NULL, (UBYTE *)
     "Gadget proporzionale mobile lungo due direzioni",
     &nomi[2]},
    {2, 0, JAM2, 290, 130, NULL,
     (UBYTE *)"Pot vert.", &nomi[3]},
    {2, 0, JAM2, 290, 140, NULL,
     (UBYTE *)"Pot orizz.", &nomi[4]}
};

struct IntuiText PotVert =
    {2, 0, JAM2, 380, 130, NULL,
     (UBYTE *)"0x0000", NULL};

struct IntuiText PotOrizz =
    {2, 0, JAM2, 380, 140, NULL,
     (UBYTE *)"0x0000", &PotVert};

struct NuovoScreen NuovoScherMoDiControllo = {
    0, 0, 640, 256, /* Coord. e dimensioni dello schermo */
    3, /* Numero di bit plane */
    (UBYTE)2, (UBYTE)1, /* Colori per lo schermo */
    HIRES, /* Modo per la ViewPort */
    CUSTOMSCREEN, /* Tipo di schermo */
    NULL, /* Fonte per lo schermo */
    (UBYTE *)"ScherMo di prova", /* Titolo dello schermo */
    NULL, /* Gadget presenti sullo schermo */
    NULL, /* Puntatore alla struttura BitMap */
};

struct NewWindow NuovaFinestraDiControllo = {
    20, 20, /* Coordinate del vertice in alto a sinistra */
    600, 160, /* Dimensioni iniziali (larghezza e altezza) */
    2, 3, /* Colori dei dettagli e dello sfondo */
    GADGETUP | GADGETDOWN | CLOSEWINDOW | MOUSEMOVE | MENUPICK,
    WINDOWDRAG | WINDOWDEPTH | WINDOWCLOSE | ACTIVATE,
    &PropGadget_free, /* Puntatore al primo gadget */
};

```

```

NULL, /* Nessun puntatore a CheckMark */
(UBYTE *)"Gadget proporzionali", /* Titolo finestra */
NULL, /* Nessun puntatore a una struttura Screen */
NULL, /* Nessun puntatore a una struttura BitMap */
20, 20, /* Dimensioni minime (non e' consentito) */
600, 160, /* m massime (il ridimensionamento) */
CUSTOMSCREEN /* Tipo di schermo */
};

.....
* Corpo principale del programma *
.....

main()
{
    ULONG Segnali, MIClass, MICode;
    APTR MIAddress;
    LONG gad_pos, real_pos;
    SHORT MouseX, MouseY, prima_stringa;
    BOOL scrivi_stringhe;

#ifdef DEBUG
    /* Controllo sulle costanti definite */
    puts("Debugging...");
    if ((STRINGHE_TOTALI > 10L) ||
        (STRINGHE_VISIBILI >= STRINGHE_TOTALI) ||
        (STRINGA_INIZIALE > (STRINGHE_TOTALI-STRINGHE_VISIBILI)))
    {
        puts("Le costanti non rientrano nei relativi range!");
        exit(FALSE);
    }
    puts("Le costanti rientrano nei relativi range");
#endif

    if (!(IntuitionBase = (struct IntuitionBase *)
        OpenLibrary("intuition.library",
        (LONG)LIBRARY_VERSION)))
    {
        puts("Non riesco ad aprire la libreria di Intuition");
        MIElimino();
        exit(FALSE);
    }

    if (!(GfxBase = (struct GfxBase *)
        OpenLibrary("graphics.library",
        (LONG)LIBRARY_VERSION)))
    {
        puts("Non riesco ad aprire la libreria grafica");
        MIElimino();
        exit(FALSE);
    }

    /* Imposta i parametri dei gadget proporzionali */
    .....
    g_prop_scroll.Flags = AUTOKNOB | FREEVERT;

    /* Calcola la dimensione della barra di scroll perche' *
     * rappresenti la relazione fra il numero di stringhe *
     * visualizzabili e quelle effettivamente visibili */
    g_prop_scroll.VertBody =
        (ULONG)(STRINGHE_VISIBILI * 0x7FFF) / STRINGHE_TOTALI;

    /* Calcola la posizione iniziale della barra di scroll *
     * in relazione alla stringa che viene visualizzata *
     * per prima. Cambiare la costante STRINGA_INIZIALE *
     * per alterare questa posizione iniziale (range: 0-5) */
    g_prop_scroll.VertPot =
        MIN(0x7FFF,
        (((ULONG)STRINGA_INIZIALE << 16) /
        (STRINGHE_TOTALI - STRINGHE_VISIBILI)));

    g_prop_fuzzy.Flags = AUTOKNOB | FREEVERT;
    g_prop_fuzzy.VertBody = 0x3000;

    g_prop_image.Flags = FREEVERT;
    g_prop_image.VertBody = 0x3000;

    g_prop_free.Flags = FREEVERT | FREEHORIZ | PROP_BORDERLESS;
    g_prop_free.VertBody = 0x7FFF;
    g_prop_free.HorizBody = 0x7FFF;

    .....
    * Chiama la routine che dispone le stringhe da visualizzare *
    * nelle strutture IntuiText. La disposizione iniziale prevede *
    * come prima stringa quella indicata dalla costante *
    * STRINGA_INIZIALE *
    .....

    AssegnaStringhe(STRINGA_INIZIALE);
    CompletaStringhe();
};

```



```

/*****
* Apre la finestra in cui vengono visualizzati i gadget */
/*****

if (!(SchermoDiControllo = (struct Screen *)
    OpenScreen(&NuovoSchermoDiControllo)))
{ puts("Non riesco ad aprire lo schermo di controllo");
  MiElimino();
  exit(FALSE);
}

NuovaFinestraDiControllo.Screen = SchermoDiControllo;

if (!(FinestraDiControllo = (struct Window *)
    OpenWindow(&NuovaFinestraDiControllo)))
{ puts("Non posso aprire la finestra di controllo");
  MiElimino();
  exit(FALSE);
}

/*****
* Scrive le stringhe nella finestra */
/*****

PrintIText(FinestraDiControllo->RPort, &nomi[5], 0L, 0L);
PrintIText(FinestraDiControllo->RPort, &PotOrizz, 0L, 0L);
PrintIText(FinestraDiControllo->RPort,
    &IntuiStr[STRINGHE_VISIBILI-1], 0L, 0L);

/*****
* Continua l'esecuzione finche' non viene premuto il *
* gadget di chiusura della finestra */
/*****

FOREVER
{
    scrivi_stringhe = FALSE;

    /* Attende un messaggio dalla finestra *
    * e lo elabora */

    Segnali = Wait(1L <<
        FinestraDiControllo->UserPort->mp_SigFit);

    while (MiOMessaggio = (struct IntuiMessage *)
        GetMsg(FinestraDiControllo->UserPort))

        /* Risponde al messaggio dopo aver ricavato le *
        * informazioni necessarie */

        {
            MIClass = MiOMessaggio->Class;
            MICode = MiOMessaggio->Code;
            MIAddress = MiOMessaggio->IAAddress;
            MouseX = MiOMessaggio->MouseX;
            MouseY = MiOMessaggio->MouseY;

            ReplyMsg(MiOMessaggio);

            /* Valuta di che tipo di messaggio si tratta */

            switch (MIClass)
            {
                case GADGETUP:
                    if (((struct Gadget *) MIAddress)->GadgetID
                        == PROP_SCROLL)
                    {
                        prima_stringa = (((STRINGHE_TOTALI -
                            STRINGHE_VISIBILI) * g_prop_scroll.VertPot)
                            + (1L << 15)) >> 16;
                        AssegnaStringhe(prima_stringa);
                        CompletaStringhe();
                        PrintIText(FinestraDiControllo->RPort,
                            &IntuiStr[STRINGHE_VISIBILI-1], 0L, 0L);
                    }
                    if (((struct Gadget *) MIAddress)->GadgetID
                        == PROP_LIBERO)
                    {
                        /* Scrive i valori di Pot nella stringa *
                        * IntuiText. Scrive a partire dal terzo *
                        * carattere, per sovrascrivere lo "0x" */

                        sprintf(&(PotVert.IText[2]), "%04x",
                            ((struct PropInfo *)
                                (PropGadget_free.SpecialInfo))->VertPot);
                        sprintf(&(PotOrizz.IText[2]), "%04x",
                            ((struct PropInfo *)
                                (PropGadget_free.SpecialInfo))->HorizPot);
                        PrintIText(FinestraDiControllo->RPort,
                            &PotOrizz, 0L, 0L);
                    }
                }
            }
        }
    }
}
break;

```

```

case GADGETDOWN:
    break;

case MOUSEMOVE:
/*****
* Usiamo un flag per ricordarci di aggiornare le *
* stringhe successivamente, cosi' non verranno *
* aggiornate a OGNI messaggio di movimento del *
* mouse (MOUSEMOVE) */
/*****

    scrivi_stringhe = TRUE;
    break;

case CLOSEWINDOW: /* Gadget di chiusura */
/* Risponde a tutti i restanti messaggi */
while (MiOMessaggio = (struct IntuiMessage *)
    GetMsg(FinestraDiControllo->UserPort))
    ReplyMsg(MiOMessaggio);
    MiElimino();
    exit(TRUE);
    break;
} /* switch */
} /* while */

/*****
* Controlla se vanno aggiornate le stringhe. *
* Nel caso, chiama la routine per la loro *
* assegnazione alle strutture IntuiText */
/*****

if (scrivi_stringhe)
{
    prima_stringa = (((STRINGHE_TOTALI -
        STRINGHE_VISIBILI) * g_prop_scroll.VertPot)
        + (1L << 15)) >> 16;
    AssegnaStringhe(prima_stringa);
    CompletaStringhe();
    PrintIText(FinestraDiControllo->RPort,
        &IntuiStr[STRINGHE_VISIBILI-1], 0L, 0L);
} /* FOREVER */
} /* main */

MiElimino()
{
    if (FinestraDiControllo) CloseWindow(FinestraDiControllo);
    if (SchermoDiControllo) CloseScreen(SchermoDiControllo);
    if (GfxBase) CloseLibrary(GfxBase);
    if (IntuitionBase) CloseLibrary(IntuitionBase);
}

/*****
* Questa routine assegna una stringa a ognuna *
* delle strutture IntuiText che saranno visualizzate */
/*****

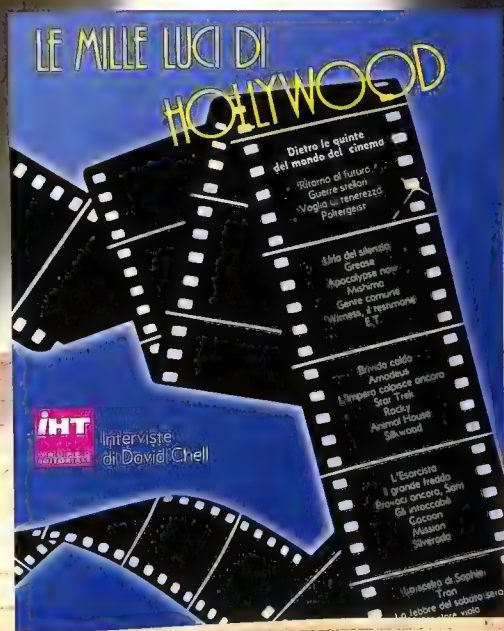
AssegnaStringhe(indice)
WORD indice;
{
    int i;
    for(i=0; i < STRINGHE_VISIBILI; i++)
        IntuiStr[i].IText = (UBYTE *)stringhe_di_testo[indice+i];
}

/*****
* Questa routine completa con degli spazi tutte le *
* stringhe che devono essere visualizzate. In *
* questo programma e' superflua, poiche' tutte le volte *
* vengono riempite di spazi le stesse stringhe. In *
* un'applicazione reale, tuttavia, le stringhe *
* potrebbero cambiare ogni volta e quindi ci sarebbe *
* la reale necessita' di ripetere l'operazione in run-time */
/*****

CompletaStringhe()
{
    int lun,i,j;
    for (i = 0; i < STRINGHE_VISIBILI; i++)
    {
        lun = strlen(IntuiStr[i].IText);
        for (j=lun; j < LUNGHEZZA_MAX - 1; j++)
        {
            *(IntuiStr[i].IText + j) = ' ';
        }
        *(IntuiStr[i].IText + LUNGHEZZA_MAX - 1) = '\0';
    }
}

```

REGALA IL CINEMA



Un'inedita e affascinante raccolta d'interviste ai più famosi "creatori di cinema" degli anni '80. Questa volta non sono al centro dell'attenzione attori e registi, ma i tecnici. Tecnici premiati con l'Oscar, esperti in campi così particolari che molti di noi non ne sospettano neppure l'esistenza. Per curiosare dietro le quinte del mondo di celluloidi. Per scoprire come si è trasformata la Hollywood degli anni '80. Per avere notizie sul futuro della computergrafica e sulle ultime novità nel campo degli effetti speciali. Ma anche per imparare ad amare il cinema e conoscerlo meglio.

AREXX E WShell, DUE UTILITY PER IL PROGRAMMATORE

Due importanti strumenti per il programmatore: il sistema AREXX, un interprete batch che ci permette di creare complessi file comandi in grado di comunicare con le applicazioni compatibili, e il pacchetto WShell, che sostituisce e migliora lo Shell del Workbench 1.3

di Eugene P. Mortimore

Presto o tardi, tutti coloro che lavorano con i computer si rendono conto che è molto comodo poter automatizzare le procedure più ripetitive. Dal momento che quasi tutti i sistemi operativi dispongono di un interprete batch, la soluzione più diffusa è la creazione di opportuni file comandi che riassumono gruppi di operazioni standard, un metodo utilizzabile anche da chi non è un programmatore.

Purtroppo si tratta di un metodo molto limitato, che consente per esempio di rendere automatica la trasformazione dei file sorgente in file eseguibili (si veda l'articolo "Un ambiente di programmazione efficiente", in questo stesso numero) ma non permette di affrontare problemi più complessi. Il problema è che generalmente i PC, come l'Amiga e gli MS-DOS compatibili, dispongono di comandi per l'interfaccia di linea insufficienti per creare file comandi molto elaborati. Inoltre, gli interpreti batch previsti dai loro sistemi operativi sono piuttosto primitivi rispetto agli interpreti per linguaggi di alto livello. I limiti più evidenti sono l'impossibilità di utilizzare nei file comandi parametri non posizionali, di creare variabili di servizio all'interno dei file (soprattutto numeriche), di eseguire calcoli, di organizzare flussi di controllo tramite cicli condizionali evoluti come while, for, repeat until, e infine di definire subroutine interne ed esterne. Insomma, l'ideale sarebbe poter redigere file comandi con strutture vicine a quelle dei linguaggi d'alto livello, come il Pascal e il Basic, ma mantenendo la filosofia d'esecuzione batch (cioè utilizzando i comandi previsti dall'interfaccia linea comando).

Un'ulteriore evoluzione dell'interprete batch potrebbe dare addirittura la possibilità di creare file comandi che diano istruzioni a un'applicazione

già in esecuzione come se fossero impartite dall'utente. Nel caso dell'Amiga, il file comandi potrebbe inviare all'applicazione una serie d'istruzioni relative a selezioni di menu e icone senza che l'utente debba muovere il mouse ed effettuarle una per una. Una situazione di questo tipo, ovviamente, esige che l'applicazione metta a disposizione una message port per ricevere le istruzioni provenienti dai file batch, e che la sondi periodicamente per verificare se sono giunti messaggi.

Il sistema AREXX è stato creato proprio per rendere possibile tutto questo anche per l'Amiga. Si tratta di un interprete batch originariamente sviluppato per il mondo IBM (in quell'ambito il suo nome è REXX), e ora trasformato e migliorato per funzionare sull'Amiga, sfruttandone le capacità multitasking e il sofisticato meccanismo di scambio dei messaggi. A questo importante pacchetto è dedicata la prima parte dell'articolo.

Nella seconda parte descriviamo invece le caratteristiche del pacchetto WShell, che rispetto allo Shell del Workbench 1.3 aggiunge diverse caratteristiche, fra cui un'interfaccia AREXX compatibile e un piping più evoluto.

Il sistema AREXX

La prima novità introdotta dal sistema AREXX è la possibilità di creare file comandi molto più complessi di quelli consentiti dall'AmigaDOS. Pur mantenendo la filosofia batch di esecuzione, con AREXX i file comandi possono contenere la maggior parte delle strutture logiche comuni a tutti i linguaggi di programmazione evoluti, come i cicli for, while, repeat until; possono impiegare subroutine e procedure, inizializzare variabili ed eseguire

calcoli numerici. In questo modo i file comandi in ambiente AREXX possono rendere automatiche non solo sequenze di comandi, ma anche scelte e cicli iterativi.

Come se non bastasse, il sistema AREXX aggiunge un'ulteriore importantissima caratteristica all'elaborazione batch: consente ai file comandi di comunicare con applicazioni dotate d'interfaccia AREXX compatibile. Un parallelo con la consueta elaborazione batch dovrebbe illustrare meglio questa possibilità. Nei file comandi ai quali siamo abituati, ogni comando viene inviato all'AmigaDOS, che provvede a eseguirlo. In altre parole, l'AmigaDOS è l'unico destinatario di tutti i comandi batch. In ambiente AREXX, invece, il file comandi può decidere a chi dev'essere inviato ogni comando.

Possiamo distinguere i comandi in due gruppi: i comandi AREXX, ovvero quelli destinati solo all'interprete, e i comandi non AREXX, ovvero quelli che intendiamo impartire a un'applicazione o al sistema operativo. I comandi AREXX sono quelli che servono per indicare a quale message port è indirizzata la linea comando, per segnalare la fine del gruppo d'istruzioni e così via. I destinatari dei messaggi si dividono a loro volta in tre categorie fondamentali: lo stesso interprete AREXX (per i comandi AREXX), l'AmigaDOS come avviene nei consueti file comandi, e infine qualunque applicazione AREXX compatibile. L'ultimo destinatario rappresenta un'importante novità, perché permette di operare su un'applicazione dall'esterno dell'applicazione stessa. Per esempio, tramite un file comandi si potrebbe selezionare una serie di voci distribuite su vari livelli di menu.

Infine, un ulteriore vantaggio di cui dispongono le applicazioni AREXX compatibili è la possibilità d'impiegare lo standard AREXX per comunicare fra loro anche se il sistema AREXX non è stato installato. Questo standard prevede in pratica che le applicazioni immettano nel sistema le loro rispettive message port AREXX, e che si scambino messaggi utilizzando una particolare struttura di dati. In questo modo possiamo creare piccole applicazioni che aumentino le prestazioni dei programmi a nostra disposizione. Se per esempio il sistema di selezione dei file su disco dell'editor *TxE*d non ci piace, possiamo creare un'applicazione che apra un requester più efficace e che poi comunichi all'editor il nome del file da caricare. Ecco un altro esempio sull'uso dello standard AREXX come sistema di comunicazione fra le applicazioni: alcuni programmatori hanno elaborato i menu dell'editor *TxE*d per inviare il contenuto di uno dei suoi buffer a un programma di comunicazione AREXX compatibile, il quale lo inoltra automaticamente lungo la linea telefonica all'utente desiderato. Con il sistema AREXX disponiamo in pratica di uno standard efficiente per far comunicare fra loro applicazioni diversissime... e non c'è nemmeno bisogno del pacchetto AREXX.

Vediamo ora come si possono sfruttare le

caratteristiche di questo sistema in un caso pratico. Supponiamo di voler realizzare un programma musicale in grado di suonare brani scritti secondo un linguaggio di nostra creazione, e che la stesura dei brani possa avvenire tramite qualunque editor. Se il programma musicale e l'editor potessero comunicare, saremmo in grado di scrivere un pezzo e di sentirne l'esecuzione senza necessariamente uscire dall'editor. Illustriamo come ci si può riuscire con l'aiuto dei file batch.

Dopo aver creato nel programma musicale un'interfaccia AREXX, lo mandiamo in esecuzione e apriamo un editor. Supponendo che si tratti di un editor AREXX compatibile in grado di aprire più file in più finestre, scriviamo in una finestra il brano musicale adottando il linguaggio che abbiamo creato e che il programma musicale è in grado d'interpretare. Quando il brano è pronto, apriamo un'altra finestra dell'editor e vi scriviamo un semplice file comandi AREXX che effettua le seguenti operazioni: ordina all'editor, tramite la sua interfaccia AREXX, di salvare in un file il contenuto della finestra dove abbiamo scritto il brano musicale, preleva tale file e lo invia al programma musicale che, ricevendolo tramite la sua interfaccia AREXX, provvede a eseguirlo. Nel caso sia stato possibile mandare in esecuzione il file comandi senza accedere al CLI, ogni operazione è stata svolta senza uscire dall'editor.

L'interfaccia batch AREXX potrebbe anche servire per creare un "filtro" fra due applicazioni, cioè un file comandi che prelevi l'output della prima, lo codifichi secondo il formato previsto dalla seconda e lo invii a quest'ultima. Modifichiamo leggermente l'esempio precedente per illustrare anche questo tipo di utilizzazione: abbiamo ancora l'editor, ma questa volta disponiamo di un programma musicale realizzato da altri, il cui linguaggio non ci risulta gradito. Possiamo allora redigere i brani musicali secondo un nostro personale linguaggio e delegare al file comandi AREXX non solo l'invio del brano al programma musicale, ma anche la sua traduzione nel linguaggio previsto dal programma. Ovviamente in questo caso il file comandi da creare è molto più complesso e l'interprete AREXX diventa assolutamente indispensabile: creare un filtro come quello descritto è assolutamente impossibile con il semplice interprete batch dell'Amiga.

Il disco del pacchetto AREXX

La Tavola 1 presenta una breve descrizione dei file più significativi presenti sul disco del pacchetto AREXX versione 1.06. Si tratta dei file dai quali dipende il meccanismo di comunicazione tra message port, i file eseguibili invocati direttamente dall'interprete e i file INCLUDE per il linguaggio C e l'Assembly che i programmatori devono impiegare per creare programmi AREXX compatibili.

Per quanto riguarda i file che non descriveremo,

Nome del file Descrizione del contenuto

File di startup contenuti nella directory radice

RxInstall	Programma per l'installazione di AREXX in un Amiga con un solo disk drive
Start-ARExx	File di startup che configura il sistema AREXX
ReadMe	File contenente estensioni al manuale per l'utente

Utility contenute nella directory c

hi	Serve a impostare a 1 il flag globale di alt di AREXX; interrompe l'esecuzione di tutti i file comandi in corso
loadLib	Carica in memoria le librerie shared relative ad AREXX e le inserisce nella lista di sistema LibList dell'Exec
rexxmast	Carica e attiva l'interprete residente AREXX
rx	Manda in esecuzione file comandi AREXX
rxcl	Disattiva l'interprete AREXX
rxlib	Carica in memoria una libreria di funzioni perché i programmi AREXX se ne possano servire
rxset	Aggiunge un clip alla lista dei clip mantenuta dall'AREXX
te	Porta a 0 il flag globale di Trace, disabilitando tale funzione
tcc	Disattiva la finestra di Trace globale, in modo che l'output prodotto dal trace di un file comandi appaia sul dispositivo di output di default
tco	Apri la finestra globale di Trace, dirigendovi l'output prodotto dal trace di tutti i file comandi AREXX
ts	Abilita il modo Trace di tutti i programmi AREXX in esecuzione
WaitForPort	Attende per dieci secondi l'apparizione di una message port nelle lista di sistema

File INCLUDE in C della directory include relativi al sistema AREXX

storage.h	File INCLUDE principale dell'AREXX, con la definizione di importanti strutture
errors.h	Definizione dei messaggi d'errore di cui fa uso l'interprete AREXX
rxslib.h	Definisce la struttura Base presente all'interno della libreria AREXX rexxsyslib.library
rexxio.h	Definisce le strutture di dati utilizzate dalle routine di I/O bufferizzato

File INCLUDE in Assembly della directory include relativi al sistema AREXX

storage.i	Analogo al file storage.h descritto sopra
errors.i	Analogo al file errors.h
rxslib.i	Analogo al file rxslib.h
rexxio.i	Analogo al file rexxio.h
rxsupplib.i	Definisce le strutture di dati usate dalla libreria AREXX rexxsupport.library

Librerie shared presenti nella directory libs

rexxsupport.library	Libreria shared di supporto all'interprete AREXX
rexxsyslib.library	Libreria shared contenente le funzioni dell'interprete

File presenti nella directory tools

hosttest	File d'esempio che mostra come creare interfacce AREXX nei propri programmi
rexttest	File che controlla il funzionamento dell'interprete AREXX quando comunica con un altro programma
rexxhost.c	Esempio, in sorgente C, di applicazione contenente una message port pubblica
dispatch.c	Sorgente C di un programma per verificare l'esecuzione delle funzioni esterne
rxexample.i	File INCLUDE Assembly necessario per utilizzare la librerie esterne personalizzate create con il tool RxExample
hosttest.asm	Listato in Assembly del programma Hosttest
loadlib.asm	Listato in Assembly del tool LoadLib, che carica in memoria librerie shared
rexttest.asm	Listato in Assembly del programma Rexttest
rx.asm	Listato del comando rx, che manda in esecuzione un file comandi
rxexample.asm	Listato del programma RxExample, che permette la creazione di librerie di funzioni AREXX

Tavola 1: elenco dei file più significativi contenuti nelle principali directory sul disco del pacchetto AREXX versione 1.06

alcuni servono a sfruttare da *Workbench* particolari caratteristiche del pacchetto AREXX, e altri (più di cinquanta) sono i comandi ARP (Amiga Replacement Project). Vi sono inoltre alcuni esempi di file comandi AREXX compatibili, file che mostrano come rendere AREXX compatibili i propri programmi in Modula 2, e infine un file che costituisce una speciale libreria matematica dedicata al sistema AREXX.

I file della directory radice. I primi file elencati nella Tavola 1 sono quelli contenuti nella directory radice. RxInstall è un programma che serve per installare il sistema AREXX su un Amiga provvisto di un solo disk drive. Start-ARExx è in pratica una startup-sequence che serve per configurare

AREXX: questo file è presentato senza alterazioni nel Listato 1.

Come si può vedere, per prima cosa carica in memoria tre librerie shared, *mathiecedoubbas.library*, *rexxsupport.library* e *rexxsyslib.library*; la prima consente l'esecuzione di calcoli in doppia precisione all'interno dei file comandi, mentre le altre due vengono impiegate solo a livello di run time. L'istruzione successiva manda in esecuzione l'interprete batch AREXX, il quale procede con la propria sequenza d'inizializzazione (*rexxmast* sta per programma AREXX "master"). A questo punto, il file comandi manda in esecuzione *WaitForPort*, il quale attende per un massimo di dieci secondi che nella lista di sistema

```

CD AREXX1.06
LOADLIB lib/mathieeedoubbas.library
LOADLIB lib/rexxsyslib.library
LOADLIB lib/rexxsupport.library
c/REXXMAST
c/WAITFORPORT REXX
c/RXLIB rexxsupport.library 0 -30 0
ASSIGN REXX: AREXX1.06:rexx

```

Listato 1: il file comandi di startup-sequence, Start-AREXX

delle message port appaia una porta di nome REXX, la porta che la procedura d'inizializzazione dell'interprete batch AREXX aggiunge al sistema tramite la funzione AddPort dell'Exec; la comparsa di questa porta assicura che l'interprete si è correttamente installato ed è pronto per operare.

In seguito il tool rxlib attiva la libreria rexxsupport.library con i seguenti parametri: priorità 0, offset della funzione Query all'interno della libreria uguale a -30, e versione richiesta della libreria uguale a 0 (significa che qualsiasi versione della libreria presente sul disco è accettabile). Il tool colloca inoltre la libreria di supporto in una lista di librerie che viene conservata dall'interprete residente AREXX.

L'ultima fase del processo di attivazione è l'assegnazione del dispositivo logico REXX: alla directory rexx contenuta nel disco sistema del sistema AREXX. È in questo dispositivo logico che l'interprete batch si aspetta di trovare i file comandi AREXX da eseguire (si noti che i nomi di questi file devono essere dotati dell'estensione ".rexx"). A questo punto l'installazione di AREXX è completata, e l'utente può iniziare a servirsene.

I file della directory c. La Tavola 1 prosegue elencando una serie di tool. LoadLib, rxlib e WaitForPort li abbiamo appena esaminati, vediamo le funzioni svolte dagli altri.

Il tool hi consente di forzare l'interruzione di tutti i file comandi AREXX in esecuzione. Il tool rx serve a far eseguire dall'interprete batch AREXX un file comandi; corrisponde quindi al comando EXECUTE dell'AmigaDOS. Il tool rxc serve a rimuovere dalla memoria l'interprete AREXX residente: dopo la sua esecuzione, nessun altro file comandi potrà essere eseguito. Il tool rxset serve per aggiungere un clip, dotato di nome proprio, alla lista dei clip mantenuta dal sistema AREXX. È in questa lista che si collocano le stringhe di caratteri a cui devono accedere i vari file comandi, ed eventualmente altre applicazioni.

I quattro tool successivi hanno a che fare con il trace dei file comandi AREXX. Il trace serve principalmente per effettuarne il debug, poiché consente di osservare l'esecuzione di un file comandi istruzione per istruzione, e permette quindi di analizzare il flusso delle operazioni e di verificare che sia quello previsto. Al termine di ogni istruzione, l'utente può esaminare lo stato delle variabili e, più in generale, rilevare che cos'è accaduto durante l'esecuzione. Si tratta di una funzione analoga ai comandi trace presenti in

strumenti di debug come il *Source Level Debugger* per il linguaggio C.

Il comando ts fa in modo che tutti i file comandi in esecuzione da parte dell'interprete AREXX entrino in modalità d'esecuzione trace, cioè vengano eseguiti in modo single-step (un passo alla volta). L'output prodotto da quest'operazione appare nelle finestre relative ai singoli file comandi.

Il comando tco apre invece una finestra di sistema verso cui viene ridiretto il trace di tutti i file comandi AREXX attivi: in questo modo non si verifica più la sovrapposizione dei messaggi di debug con il normale output del file comandi in esecuzione "controllata". Con il comando tcc si chiude questa finestra, ritornando alla condizione in cui i prompt del file comandi in esame si mischiano con quelli generati dalle operazioni di debug.

L'utility te, infine, disabilita il modo trace per tutti i file comandi AREXX attivi, facendo continuare la loro esecuzione a velocità normale.

I file della directory include. I file successivi vengono usati per lo più da chi sviluppa programmi in C o in Assembly, e desidera includere al loro interno un'interfaccia AREXX, cioè una message port configurata secondo le specifiche dello standard AREXX. Il primo gruppo di file è destinata al C, e contiene la definizione delle strutture e delle costanti che occorre usare per interfacciare correttamente un'applicazione con l'interprete AREXX.

Il secondo gruppo è invece dedicato a chi lavora in linguaggio Assembly, e contiene la traduzione in questo linguaggio delle definizioni contenute negli altri file INCLUDE.

I file della directory libs. Questi file sono, come indica il nome della directory, particolari librerie shared previste dal sistema AREXX, che vanno ad aggiungersi a quelle standard previste dall'Amiga. Ricordiamo a questo proposito che le librerie shared sono librerie che vengono aperte dai task in fase di run time per accedere alle funzioni in esse contenute (si noti che l'apertura di una libreria shared può comportare il suo caricamento da disco qualora non sia residente e non sia già stata caricata). La prima libreria citata nella tavola è rexxsupport.library, nella quale sono contenute funzioni di supporto per l'interprete, mentre la seconda, rexxsyslib.library, contiene funzioni di base.

I file della directory tools. Tra tutti i file contenuti in questa directory, la Tavola 1 riporta quelli realizzati appositamente per far sì che un'applicazione diventi AREXX compatibile. Si tratta di due tool eseguibili, due sorgenti in C, un file INCLUDE Assembly, e cinque programmi sorgente in Assembly.

Il primo tool, hostest, è il più importante: serve come test per verificare che il colloquio tra un'applicazione e l'interprete AREXX avvenga correttamente. In pratica, se si è scritto un programma che crea una message port pubblica e

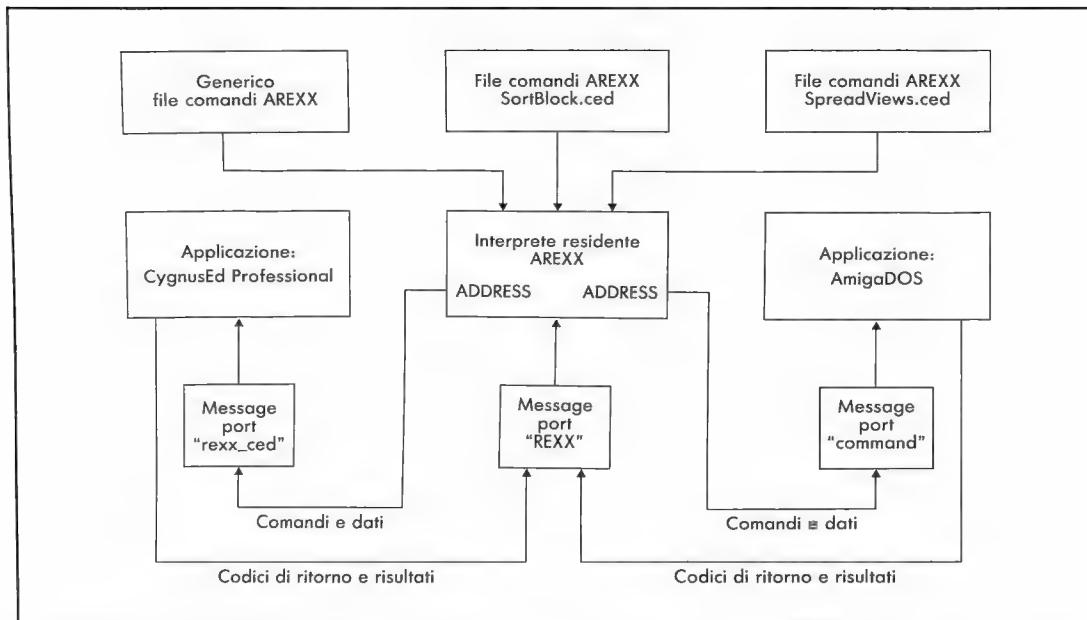


Figura 1: il programma AREXX controlla il funzionamento di un editor di testi e invia comandi all'AmigaDOS

successivamente la utilizza per comunicare con l'interprete residente (a patto che questo sia già stato installato con rexxmast), attraverso il programma hosttest è possibile verificare che tutto funzionerà correttamente; il codice sorgente Assembly di tale programma si trova nel file hosttest.asm.

L'altro programma, rexxtest, è stato progettato per controllare se eseguendo un file comandi AREXX si riesce a comunicare con un'applicazione; il codice sorgente Assembly si trova nel file rexxtest.asm.

AREXX e le applicazioni compatibili: CygnusEd Professional

La Figura 1 illustra un possibile uso del sistema AREXX per mettere in comunicazione due ambienti diversi. Nell'esempio, i due ambienti sono rispettivamente l'editor *CygnusEd Professional*, che è AREXX compatibile, e l'AmigaDOS; tramite un opportuno file comandi AREXX, viene ordinata l'esecuzione di particolari operazioni senza che l'utente debba interagire con i menu dell'editor o debba aprire una finestra CLI per impartire comandi all'AmigaDOS.

L'editor *CygnusEd* installa nel sistema una message port pubblica denominata "rexx_ced": i task possono individuarla utilizzando il suo nome (che ovviamente devono conoscere), e quindi possono indirizzarla per inviare messaggi all'editor. Anche i file comandi AREXX destinati all'editor devono impiegare la message port rexx_ced. Analizziamo ora come viene aperto un canale di

comunicazione con l'AmigaDOS.

Il rettangolo sulla destra presente nella Figura 1 rappresenta l'AmigaDOS, che come abbiamo già spiegato è il secondo tipo di destinatario dei comandi batch AREXX, ma in generale potrebbe trattarsi di qualsiasi altra applicazione.

L'AmigaDOS non nasce AREXX compatibile, e quindi non dispone di un'apposita interfaccia. Ciò nonostante, nella filosofia AREXX è importante che il file comandi lo veda come vedrebbe qualsiasi altra applicazione. Perché questo avvenga, AREXX finge che l'AmigaDOS possieda una message port di nome "command", cioè chiede di utilizzare questo nome quando si desidera che una serie di comandi venga indirizzata all'AmigaDOS (si noti che il nome "command" nell'originale REXX per gli IBM indirizza il sistema operativo). Ovviamente, questa message port non è in realtà controllata dall'AmigaDOS, ma da un processo appositamente predisposto per interfacciare l'AmigaDOS con il sistema AREXX. Per quanto riguarda l'utente, comunque, la tecnica usata è irrilevante.

Vediamo dunque come dovrebbe essere scritta un'istruzione in un file comandi AREXX diretto all'AmigaDOS. Se si desidera per esempio cancellare dal disco il file "Archivio", l'istruzione è:

```
ADDRESS command "DELETE Archivio"
```

Questa istruzione, diretta alla porta "command", in realtà fa in modo che il comando "DELETE Archivio" giunga all'AmigaDOS come se fosse stato impartito da una finestra CLI o dallo Shell. ADDRESS è un comando AREXX adibito a selezionare la message port dell'applicazione alla

quale devono essere inviati i comandi. Tutti i comandi non AREXX che lo seguono verranno inviati a quella message port fino a quando non ne verrà indicata un'altra.

L'interprete residente

Il terzo rettangolo della Figura 1, quello centrale, rappresenta l'interprete residente AREXX, cioè l'interprete che esegue i file comandi AREXX (simbolicamente rappresentati dai rettangoli sovrastanti). L'utente può avviarne l'esecuzione tramite il comando rx, il quale sottopone all'interprete il file comandi indicato come argomento.

L'interprete considera ogni file comandi come un distinto processo AmigaDOS, e invia i comandi non AREXX all'applicazione che è stata selezionata dall'ultimo ADDRESS. Inoltre si aspetta di ricevere nella sua message port pubblica "REXX" le risposte dalle applicazioni che a mano a mano indirizza. Tutte le applicazioni che contengono un'interfaccia AREXX compatibile conoscono il nome di questa message port pubblica, reperibile all'interno della lista di sistema PortList.

Ovviamente, sulla falsariga della Figura 1, è possibile inserire un numero qualsiasi di applicazioni AREXX compatibili e un numero qualsiasi di file comandi da eseguire. Nei due esempi che seguono, il flusso logico illustrato dalla Figura 1 viene mostrato sotto un profilo più pratico.

Primo esempio: ordinamento di un blocco di testo

Il Listato 2 riporta un esempio abbastanza tipico di file comandi AREXX. Si tratta di un "programma" di nome SortBlock.ced contenuto nel disco dell'applicazione *CygnusEd*; lo illustriamo soprattutto perché consente di vedere da vicino come con il sistema AREXX è possibile comandare dall'esterno un'applicazione, e ne traduciamo in italiano i prompt per l'utente. SortBlock.ced ha lo scopo d'interagire con l'editor *CygnusEd* al fine di copiare in un file su RAM disk un blocco di testo selezionato in precedenza, di ordinarlo alfabeticamente attraverso il comando SORT dell'AmigaDOS, e infine di reinserirlo nella finestra dell'editor da cui è stato temporaneamente estratto, il tutto senza alcun intervento diretto da parte dell'utente.

La prima riga esegue l'istruzione AREXX OPTIONS e indica la keyword RESULTS, la quale ordina all'interprete di attendere una risposta ogni volta che invia un messaggio a una message port; l'interprete poi memorizzerà la risposta, di solito una stringa, nella variabile result.

L'istruzione seguente, ADDRESS, indica all'interprete che i successivi comandi (salvo i comandi AREXX) dovranno essere inviati a una message port pubblica di nome "rexx_ced": in questo modo viene in pratica creato un canale di comunicazione tra la porta di nome "REXX" dell'interprete e la

```
OPTIONS RESULTS
ADDRESS "rexx_ced"
cut block
IF result = "RESULT" THEN DO
  save block "ram:tmp$sortblock"
  ADDRESS command "SORT RAM:tmp$sortblock RAM:tmp
  $sorted"
  ADDRESS "rexx_ced" "include file RAM:tmp$sorted"
  status 21
  ADDRESS command "DELETE RAM:tmp$sortblock"
  DELETE RAM:tmp$sorted
END
ELSE
  okay1 "Non posso procedere. Nessun blocco
  selezionato."
EXIT
```

Listato 2: il file comandi SortBlock.ced contenuto in *CygnusEd*

porta dell'applicazione *CygnusEd*, che rimarrà aperto fino a quando l'interprete non troverà un'altra istruzione ADDRESS.

La linea successiva invia il comando cut block all'editor, con il quale il blocco di testo che l'utente ha selezionato (con l'opzione SetMark) viene trasferito in un buffer interno, il buffer Paste dell'editor. Questo comando, che l'editor ovviamente riconosce attraverso la sua interfaccia AREXX, corrisponde nei menu alla voce Cut block: aver inviato il comando cut block dall'interfaccia AREXX dell'editor ha lo stesso effetto di una selezione della voce Cut block del menu da parte dell'utente.

Con la successiva linea comandi, viene controllato l'esito che l'editor restituisce dopo aver eseguito il comando cut: se la variabile result contiene la stringa "RESULT", cioè il nome della variabile stessa, significa che non è stato restituito alcun codice d'errore, e che dunque l'operazione è stata portata a termine con successo. Questo test sfrutta una particolarità dell'interprete AREXX: ogni variabile alla quale non è ancora stato assegnato un valore assume il valore stringa del proprio nome.

Dal momento che il blocco è stato trasferito con successo, il file comandi prosegue dicendo all'editor di copiarlo in un file temporaneo su RAM disk, di nome "tmp\$sortblock". Anche in questo caso, il comando save block è uno di quelli riconosciuti dall'editor.

Successivamente, viene indirizzato l'AmigaDOS e gli viene inviato il comando SORT, indicando come argomenti il nome del file da ordinare e il nome del file nel quale dev'essere memorizzato il testo ordinato ("RAM:tmp\$sorted"). Come abbiamo già spiegato, per comunicare con l'AmigaDOS occorre indirizzare la porta di nome command. A questo punto il blocco di testo è stato ordinato ed è pronto per essere reinserito nel testo all'interno dell'editor dal quale è stato estratto: per questo scopo viene impiegato il comando include che *CygnusEd* riconosce attraverso la sua interfaccia AREXX.

Il successivo comando, status 21, viene inviato

automaticamente alla porta "rexx_ced" dell'editor, dal momento che l'ultimo comando ADDRESS ha indirizzato proprio quella porta. Il comando status è anch'esso un comando riconosciuto da *CygnusEd*, come spiegano i manuali, e serve a ottenere informazioni riguardanti il numero delle finestre aperte, le dimensioni della finestra corrente e così via. In particolare, indicando come argomento il codice 21 si ottiene nella variabile result il nome per esteso del file in corso di elaborazione all'interno della finestra di *CygnusEd*.

Le ultime due linee comando di questo file indirizzano nuovamente l'AmigaDOS e gli ordinano di cancellare i due file temporanei RAM:tmp\$sortblock e RAM:tmp\$sorted. Quando anche queste due operazioni sono terminate, la parola chiave END chiude il costrutto condizionale IF THEN.

L'ELSE che segue viene eseguito solo se *CygnusEd* ha rilevato una condizione d'errore eseguendo il comando cut block, ad esempio per la mancanza di un blocco di testo selezionato. In questo caso si fa visualizzare a *CygnusEd* un requester (tramite il comando okay1) per avvisare l'utente che non è stato selezionato alcun blocco, e che quindi non è possibile effettuare l'ordinamento. L'ultima istruzione del file, EXIT, è facoltativa e segnala che il file comandi è concluso.

Secondo esempio: un intervento sulle finestre

Il Listato 3 illustra un secondo esempio di file comandi AREXX, SpreadViews.ced, anch'esso presente nel disco dell'editor *CygnusEd*. Come nell'esempio precedente, i messaggi per l'utente sono stati tradotti in italiano. Analizziamolo.

Questo programma serve a rendere uguali (verticalmente) tutte le finestre aperte dell'editor *CygnusEd*, in modo che contengano lo stesso numero di linee di testo. Può servire se, nella stesura di un programma in C, si sta lavorando su diversi file e viste di file, e procedendo con il lavoro le finestre relative alle varie viste sono mutate di dimensioni. In tal caso diventa comodo poterle riportare a una dimensione verticale unica per seguire con più ordine l'andamento del programma.

Prima di procedere occorre ricordare che le finestre che si possono aprire in *CygnusEd* non sono le consuete finestre di Intuition (spostabili con il mouse, ridimensionabili e sovrapponibili): si tratta di semplici suddivisioni orizzontali dello schermo in più aree, nelle quali possono apparire parti diverse dello stesso testo o addirittura testi provenienti da file diversi. Pertanto la loro gestione avviene esclusivamente tramite i menu, e non si possono sovrapporre, in quanto dividono semplicemente la finestra occupata dall'editor sullo schermo. Nella spiegazione che segue ci riferiremo a loro chiamandole finestre, ma non devono essere confuse con le finestre di Intuition

```

OPTIONS RESULTS
status 54
numlines = ( result - 3 ) % 8
status 66
numviews = result
IF numviews = 1 THEN DO
  okay1 "E' presente una sola finestra!"
  EXIT
  END
  desire = numlines % numviews
  remainder = (numlines - (desire * numviews))
  desire = desire - 1
  curview = 0
  status 49
  maxy = result
  newres = maxy
  DO UNTIL maxy > newres
    maxy = newres
    next view
    curview = curview + 1
    status 49
    newres = result
  END
  previous view
  curview = curview - 1
  DO WHILE numviews > 1
    status 51
    origlines = result
    diff = origlines - desire
    IF remainder > 0 THEN DO
      diff = diff - 1
      remainder = remainder - 1
    END
    DO WHILE diff < 0
      grow view
      diff = diff + 1
    END
    DO WHILE diff > 0
      shrink view
      diff = diff - 1
    END
    previous view
    curview = curview - 1
    numviews = numviews - 1
  END
  DO WHILE curview > 0
    next view
    curview = curview + 1
  END
  EXIT

```

Listato 3: SpreadViews.ced, un esempio di file comandi AREXX

con le quali siamo abituati a interagire. Questa premessa è fondamentale per comprendere il funzionamento del file comandi SpreadViews.ced.

In questo file comandi vengono utilizzati diversi comandi status, calcoli aritmetici e variabili non soggette a particolari tipi (stringa, numeriche...). Anziché analizzare ogni singola linea di programma, preferiamo metterle in evidenza gli aspetti chiave, che saranno utili per comprendere alcune peculiarità del linguaggio AREXX.

Il flusso delle operazioni compiute dal file comandi è il seguente: viene calcolata l'altezza in righe di testo della finestra principale occupata da *CygnusEd* sullo schermo e successivamente viene compiuto un test di uscita sul numero di finestre

aperte; se viene indicata una sola finestra, un requester avvisa l'utente dell'anomalia e viene conclusa l'esecuzione del file comandi. Se ci sono due o più finestre aperte, viene calcolata la dimensione verticale (desire) che le finestre devono assumere per dividere lo spazio verticale disponibile in parti uguali. In questo calcolo si tiene conto del fatto che dividendo l'altezza disponibile per il numero di finestre aperte non è detto che si ottenga un risultato intero, cioè può darsi che si ottenga uno scarto di un certo numero di righe che il file comandi inserisce nella finestra più bassa.

Successivamente, viene selezionata la finestra più bassa per entrare nel ciclo di ridimensionamento delle finestre con la finestra disposta più in basso come finestra attiva. Il comando next view dell'editor seleziona sempre la finestra che sullo schermo appare appena più in basso di quella attuale, e riprende circolarmente da quella più alta quando la finestra selezionata è quella più in basso. Durante questo ciclo viene incrementata la variabile curview, che alla fine consentirà al file comandi di rifelezionare la finestra che risultava attiva quando era stato mandato in esecuzione.

A questo punto la finestra attiva è quella che sullo schermo occupa la parte più bassa. Il file comandi entra in un ciclo che modifica l'altezza di ogni finestra con il valore contenuto in desire, partendo dalla finestra attiva e selezionando tutte le altre finestre dal basso verso l'alto. In questo ciclo troviamo un test condizionale che verifica se ci sono righe in eccesso, nel qual caso provvede ad aumentare opportunamente le dimensioni della prima finestra e ad azzerare lo scarto di modo che con le finestre successive questa compensazione non venga più effettuata. Quando il ciclo è terminato, tutte le finestre aperte sullo schermo, tranne eventualmente quella più in basso, appaiono della stessa altezza.

L'ultimo ciclo, infine, fa in modo che venga rifelezionata la finestra che risultava attiva prima che iniziasse l'esecuzione del file comandi. Per identificarla, il ciclo sfrutta il valore contenuto nella variabile curview.

Descriviamo ora le principali variabili impiegate nel file comandi. Il comando status 54 restituisce l'altezza in pixel della finestra impiegata da *CygnusEd*: questa dimensione viene considerata l'altezza massima per una finestra. Nel programma viene diminuita di tre e poi divisa per otto al fine di esprimerla come numero di righe di testo; il risultato di questa operazione viene memorizzato nella variabile numlines. Il comando status 66 restituisce il numero totale di finestre dell'editor attualmente aperte, che viene memorizzato nella variabile numviews; quest'ultima informazione, insieme con la precedente, viene usata per calcolare la dimensione verticale alla quale occorre uniformare tutte le finestre aperte. Questa dimensione viene memorizzata nella variabile desire, ed è il risultato arrotondato di una divisione; con

l'istruzione successiva si calcola il resto della divisione e lo si inserisce nella variabile remainder, che verrà consultata per decidere quante righe aggiungere alla finestra più in basso. Il comando status 49 serve per conoscere la coordinata y del vertice in alto a sinistra della finestra corrente, l'ultima che è stata selezionata dall'utente prima di mandare in esecuzione il file comandi; questo valore viene inizialmente memorizzato nella variabile maxy, mentre nel successivo ciclo DO UNTIL viene ripetutamente memorizzato nella variabile newres. Il comando status 51 restituisce il numero di righe di testo che possono essere visualizzate all'interno della finestra corrente, il che è utile per sapere di quanto vanno modificate le dimensioni delle varie finestre; questo dato viene memorizzato nella variabile origlines. Infine, data una finestra, la variabile diff rappresenta la differenza fra la sua altezza attuale e quella che dovrà assumere; se remainder indica la presenza di uno scarto, diff viene opportunamente diminuita per compensare il fatto che il numero di righe disponibile non è un multiplo esatto del numero di finestre aperte. In seguito viene impiegata per aumentare o diminuire l'altezza della finestra corrente. Si noti che per compiere il ridimensionamento vengono impiegati rispettivamente il comando grow view e shrink view. Si noti inoltre che questi due comandi, come anche next view e previous view, in *CygnusEd* corrispondono a voci di particolari menu che il file comandi attiva attraverso l'interfaccia AREXX.

Come ci mostra anche il file comandi SpreadViews.ced, AREXX permette di creare flussi logici e di eseguire calcoli aritmetici di sufficiente complessità per svolgere compiti che normalmente affidiamo a linguaggi di programmazione, e non a file comandi eseguiti in batch. In particolare, l'inedita caratteristica che permette ai file comandi di dare istruzioni ad applicazioni AREXX compatibili si rivela davvero utile per automatizzare procedure anche molto complesse come quella che abbiamo appena descritto.

Il pacchetto WShell

Fino alla versione 1.2 del *Workbench* l'unica interfaccia linea comando a disposizione dell'utente era il famoso CLI, da sempre accusato di non reggere il confronto con la macchina per cui era stato realizzato. Inferiore perfino all'interprete linea comando MS-DOS, il CLI è in effetti il minimo di cui si possa disporre per digitare comandi dell'AmigaDOS, ma le colpe non sono tutte sue. Il CLI è solo un programma che si preoccupa di ricevere stringhe di testo dall'utente e di passarle all'AmigaDOS, il quale cerca di eseguirle aspettandosi che siano dei comandi. Per gestire le sue operazioni di I/O con l'utente, il CLI si serve per default del dispositivo logico CON: un handler che apre una finestra e svolge il ruolo d'interfaccia fra il CLI e l'utente. La vergognosa mancanza dell'edit

Nome del file	Descrizione
---------------	-------------

File contenuti nella directory radice

Install-WShell	File comandi per l'installazione di <i>WShell</i> sul disco sistema
ReadMe	File contenente informazioni su <i>WShell</i>

File della directory extras

EditMode	Tool che visualizza come parte del prompt del CLI lo stato di "inserisci/sovrascrivi" della tastiera
IsRexx	Tool per determinare se <i>AREXX</i> è attivo
LonelyShell	Serve a rilevare se è attiva una sola finestra <i>WShell</i>
WaitForPort	Analoga all'omonima funzione della Tavola 1, attende che venga creata una particolare message port
CdTitle	Serve a mostrare il nome della directory corrente nella barra titolo della finestra di <i>WShell</i>
PatchDOS	Modifica il comando AmigaDOS EXECUTE affinché funzioni correttamente con il <i>WShell</i>

File della directory sources

Alias	Comando che imposta gli alias (sinonimi)
History	Comando per la gestione del file "history" dei comandi immessi nel buffer storico
NewWSH	Aprire una nuova finestra <i>WShell</i>
RunWSH	Aprire una nuova finestra <i>WShell</i> per l'esecuzione di programmi in background (analogo al comando AmigaDOS RUN)
LoadLib	Comando per il caricamento in memoria delle librerie shared
Resi	Comando che trasforma un comando in residente

File della directory s

LoadResi	Esempio di startup-sequence che rende residenti un certo numero di comandi
Startup1	Esempio di startup-sequence che attiva l'handler ConMan e manda in esecuzione il file comandi Startup2
Startup2	Altro esempio di startup-sequence che carica alcuni comandi rendendoli residenti su un disco virtuale chiamato VDO:
Startup-Sequence	Esempio di file di startup che apre una finestra <i>WShell</i> e rende residenti i comandi RESI e ALIAS

File della directory filters

head.rexx	Un programma <i>AREXX</i> utile per disporre di una pipe attraverso la quale passano solo le prime dieci righe di caratteri
tee.rexx	Un programma <i>AREXX</i> che consente l'impiego simultaneo della console e di un altro canale come dispositivi di output

File della directory libs

WShell.library	La prima delle due librerie shared usate dal pacchetto <i>WShell</i>
ConHandler.library	La versione 1.1 della libreria shared relativa all'handler ConMan utilizzata da <i>WShell</i>

Tavola 2: elenco e descrizione dei principali file contenuti nell'interessante pacchetto per il programmatore *WShell* (versione 1.0)

di linea regolarmente imputata al CLI, per fare un esempio, è in effetti una mancanza del dispositivo logico CON:. Se si dispone del *Workbench 1.3*, si può dimostrare questa tesi digitando il comando NEWCLI NEWCON:0/0/640/200/. Appare una finestra CLI, ma come per magia disponiamo ora dell'edit di linea e perfino di un buffer storico che permette d'impartire i comandi già digitati senza riscriverli. Questo esempio mostra che esiste una ripartizione di compiti fra l'interfaccia linea comando e il dispositivo logico, o handler, impiegato come finestra di I/O.

Le vere mancanze del CLI sono per esempio l'impossibilità di definire alias, di rendere residenti i comandi, di definire un prompt più elaborato che indichi anche il cammino di ricerca seguito fino a quel momento. A cancellare con un colpo di spugna anche queste limitazioni è giunto finalmente lo Shell, l'interfaccia linea comando messa a disposizione dal *Workbench 1.3*. In pratica, con la versione 1.3 del disco sistema disponiamo ora di un'interfaccia linea comando più elaborata e di un dispositivo logico di console anch'esso migliore, un binomio che ci consente finalmente di lavorare con l'AmigaDOS in maniera spedita e intelligente.

Ma nonostante questo miglioramento, lo Shell e il dispositivo logico NEWCON: non si possono ancora definire strumenti di lavoro veramente

completi. Viste le potenzialità del sistema *AREXX* ci piacerebbe per esempio che lo Shell fosse *AREXX* compatibile, e che il piping fosse più semplice da gestire e più completo.

In nostro aiuto vengono allora il tool *WShell* e il nuovo handler ConMan, accomunati in un pacchetto che contiene anche altri tool utili per rendere più efficiente il lavoro dell'utente con l'AmigaDOS. Dal momento che alcune delle caratteristiche del pacchetto *WShell* sono comuni anche allo Shell, su di esse sorvoleremo approfondendo invece le novità.

Il pacchetto prevede un solo disco, sul quale sono presenti sia il tool *WShell* sia l'handler ConMan. Nella Tavola 2 sono elencati i significati di alcuni file, quelli più importanti. Dei rimanenti possiamo dire quanto segue.

- Nella directory rexx sono presenti nove file comandi *AREXX* d'esempio, utili per vedere come interfacciare i file comandi *AREXX* con programmi che fanno uso di specifiche caratteristiche del pacchetto *WShell*; questi file comandi sono ovviamente stampabili per essere studiati e approfonditi.

- Nella directory sources sono contenuti sette file sorgente in Assembly con lo stesso scopo dei precedenti (due sono mostrati nella tavola). Possono essere impiegati per modificare ed estendere il

tool WShell e l'handler ConMan.

L'handler ConMan differisce da NEWCON: del *Workbench 1.3* solo per la sua capacità di salvare il contenuto del buffer storico su disco e all'occorrenza di ricaricarlo.

Il tool WShell differisce dallo Shell sotto diversi aspetti. Gli alias, o sinonimi, nello Shell sono ottenuti tramite un comando residente all'interno dello stesso Shell, mentre in WShell si ottengono tramite un comando che risiede sul disco. Inoltre, il comando ALIAS previsto da WShell ha una sintassi diversa rispetto a quella prevista dallo Shell. Facciamo un esempio. Se si conta di utilizzare molte volte un comando come:

```
COPY myfile1 TO myfile2
```

si può creare un sinonimo, ottenere cioè che le stesse funzioni vengano svolte da un comando più semplice, come c12.

```
ALIAS c12=COPY myfile1 TO myfile2
```

Il piping

Questa caratteristica consente di dirigere l'output prodotto da un programma direttamente in input a un altro: si tratta in un certo senso di un'estensione che WShell aggiunge alla possibilità di ridirezione dell'output ottenuta nell'AmigaDOS con il carattere ">" o con il dispositivo logico PIPE: disponibile con la versione 1.3 del *Workbench*. A seguito di ciò, se si hanno diversi programmi concatenati, nei quali cioè l'output dell'uno deve servire come input per l'altro, è possibile generare una pipe per incanalare opportunamente i flussi di output.

Per esempio, se si desidera leggere la directory del disco e poi riordinarla alfabeticamente, magari attraverso il comando SORT dell'AmigaDOS, si possono collegare i due comandi in questo modo:

```
DIR df0: | SORT (opzioni)
```

Il carattere "|" è l'operatore che effettua la pipe in ambito WShell: in questo caso "devia" verso il comando SORT i caratteri che normalmente verrebbero stampati sullo schermo dal comando DIR. L'unico output che sarà visibile, dunque, sarà quello del SORT, dopo l'elaborazione dei dati. Questo modo di effettuare il piping si rivela molto più utile di quanto non lo sia il dispositivo logico PIPE:, e molto più simile a quello esistente in ambiente MS-DOS o Unix.

L'interfacciamento AREXX-WShell

WShell mette anche a disposizione un'interfaccia AREXX incorporata per permettere a file comandi AREXX compatibili di controllarlo dall'esterno. La presenza di quest'interfaccia AREXX consente di dare istruzioni al tool WShell anche dall'interno di un proprio programma, applicando le particolari convenzioni previste dal sistema AREXX. Quando viene impartito un comando NEWWSH, il nuovo

WShell che viene installato in memoria crea una message port AREXX compatibile e la rende pubblica, in modo che il sistema AREXX e le altre applicazioni possano comunicare con questa porta e quindi con il relativo WShell. In questo modo si riescono a eseguire comandi dell'AmigaDOS in un modo simile a quello mostrato nel Listato 1.

Sul disco del pacchetto WShell sono presenti anche i comandi ARP (Amiga Replacement Project) che sostituiscono i comandi dell'AmigaDOS contenuti nella directory c del *Workbench*. Questi comandi corrispondono a quelli forniti dalla Commodore, come COPY, INFO e MAKE-DIR, ma offrono ognuno qualcosa in più rispetto alle vecchie versioni.

In genere hanno piccoli miglioramenti o file eseguibili di dimensioni minori oppure ancora opzioni più utili di quelle disponibili con i vecchi comandi. La documentazione relativa a questi nuovi comandi si trova nella directory "ARP UserDocs" ed è simile a quella disponibile nel *Manuale dell'AmigaDOS*. Pur non essendo d'importanza essenziale all'interno del pacchetto WShell, questi nuovi comandi rappresentano comunque qualcosa in più rispetto a quanto è messo a disposizione dalla Commodore.

Il sistema AREXX e il pacchetto WShell sono venduti separatamente e sono reperibili presso:

William S. Hawes
(AREXX, \$49.95; WShell, \$49.95)
P.O. Box 308
Maynard, MA 01754, USA
(Tel. 001/617/5688695)

Entrambi i programmi funzionano su tutti i modelli dell'Amiga, con le versioni del sistema operativo 1.2 e 1.3.

La prossima puntata

Nel prossimo numero analizzeremo le tecniche di programmazione e illustreremo un sistema per quantificare il tempo d'esecuzione dei programmi. La temporizzazione degli eventi e il loro controllo nell'ambito del singolo task o dell'intero sistema ha molta importanza in una macchina multitasking come l'Amiga, dal momento che per mantenere l'efficienza del sistema i cicli della CPU non devono mai essere sprecati in rovinosi loop d'attesa.

L'articolo proporrà una panoramica su questi aspetti e presenterà alcune procedure per la misurazione dei tempi impiegati a molteplici livelli, suddividendole per aree d'interesse.

Si parlerà di come usare il sistema delle message port dell'Exec per controllare le temporizzazioni di un task, di come va usato il dispositivo Timer per ottenere risultati di precisione, e di come l'hardware intervenga in queste operazioni. Vedremo cosa accade dal punto di vista di Intuition e dell'AmigaDOS e naturalmente faremo uso di AREXX per il controllo delle temporizzazioni. ■

L'AMIGA BASIC ALLA CONQUISTA DI INTUITION: SCHERMI E FINESTRE

Grazie a Intuition, da Basic si può risparmiare memoria chiudendo lo schermo Workbench, e ridirigere l'output su uno schermo personalizzato. Sempre da Basic analizziamo anche la gestione delle finestre di Intuition

di Filippo Moretti e Dario Zanon

Con questo articolo ci proponiamo d'introdurre i lettori all'uso delle funzioni grafiche di Intuition dall'interno dei propri programmi Basic, senza far uso di moduli esterni in C o in Assembly. La libreria Intuition è ricca di funzioni per la grafica e per l'interfaccia utente, che sono in grado di soddisfare praticamente qualsiasi esigenza. Per il momento abbiamo scelto di trattare la gestione degli schermi e delle finestre.

Prima di avvicinarci all'argomento da un punto di vista pratico affrontiamo qualche nozione teorica di base. Per comunicare con il sistema operativo dell'Amiga, i programmi devono generalmente organizzare sotto forma di strutture (insiemi standard di parametri che il sistema è in grado d'interpretare) i dati che intendono inviare. Questo importante concetto vale per qualsiasi linguaggio, e quindi anche per il Basic.

Intuition prevede molte strutture (RastPort, ViewPort, Screen...), ma in questo contesto quella che più ci interessa è NewScreen.

La struttura NewScreen

La struttura NewScreen viene impiegata per aprire un nuovo schermo di Intuition. Ricordiamo che gli schermi di Intuition sono quelli che possiamo muovere e sovrapporre l'uno all'altro agendo con il mouse. Ognuno ha caratteristiche proprie indipendenti dagli altri. Lo schermo Workbench viene aperto per default dal sistema, ma i programmi ne possono aprire altri secondo le loro necessità (si ricordi che lo schermo Workbench non è lo schermo del Workbench, ma lo schermo sul quale il Workbench disegna le proprie icone; il sistema lo apre per default anche se il Workbench

non viene mandato in esecuzione, e gli assegna il nome "Workbench Screen").

Questi schermi possono essere dotati di gadget di profondità, con i quali cambiare il loro ordine di sovrapposizione, esattamente come accade per le finestre di Intuition. Quando uno schermo non serve più, può essere chiuso per liberare memoria. In questo articolo vedremo com'è possibile chiudere e riaprire anche lo schermo Workbench oltre ai normali schermi personalizzati.

Sebbene ogni schermo sia definito in memoria da una struttura Screen, i programmi che desiderano aprire nuovi schermi devono servirsi della struttura NewScreen. L'apertura di ogni nuovo schermo avviene infatti chiamando la funzione OpenScreen di Intuition e indicando l'indirizzo di una struttura NewScreen come argomento; questa funzione apre il nuovo schermo e crea una struttura Screen di cui restituisce l'indirizzo.

La struttura NewScreen è definita come segue.

```
struct NewScreen
{
    SHORT LeftEdge, TopEdge, Width, Height, Depth;
    UBYTE DetailPen, BlockPen;
    USHORT ViewModes;
    USHORT Type;
    struct TestAttr #Font;
    UBYTE #DefaultTitle;
    struct Gadget #Gadgets;
    struct BitMap #CustomBitMap;
};
```

In breve, i significati dei suoi parametri sono i seguenti. LeftEdge e TopEdge indicano le coordinate (x,y) dell'angolo superiore sinistro dello schermo. Width e Height indicano la larghezza e l'altezza dello schermo, mentre Depth indica il numero di bit plane di cui lo schermo è composto (si ricordi che il numero di bit plane di uno

VALORI CHE PUÒ ASSUMERE IL PARAMETRO VIEWMODE

Definizione C	Esadec.	Decim.	Descrizione
HIRES	0x8000	-32768	risoluzione orizzontale di 640 pixel
LACE	0x0004	4	risoluzione verticale di 512 pixel (interlace)
SPRITES	0x4000	16384	lo schermo supporta gli sprite
DUALPF	0x0400	1024	dual playfield mode (schermi sovrapposti con effetti di trasparenza)
HAM	0x0800	2048	hold and modify mode (4096 colori contemporaneamente con 6 bit plane per una risoluzione orizzontale di 320 pixel)
EXTRA_HALFBRITE	0x0080	128	extra half brite mode (512 colori contemporaneamente con 6 bit plane)

VALORI CHE PUÒ ASSUMERE IL PARAMETRO TYPE

Definizione C	Esadec.	Decim.	Descrizione
CUSTOMSCREEN	0x000F	15	schermo personalizzato (custom)
CUSTOMBITMAP	0x0040	64	indica che lo schermo dev'essere aperto con una bit map indicata dal task
SCREENQUIET	0x0100	256	non visualizza la barra titolo

Tavola 1

schermo determina la quantità di colori che vi si possono visualizzare). DetailPen e BlockPen devono contenere rispettivamente il colore del tratto e il colore di riempimento che il sistema usa per visualizzare la barra titolo e i gadget dello schermo. Type indica il tipo di schermo; Font la fonte-carattere di default dello schermo (se si indica NULL viene impiegata Topaz 8, quella residente); DefaultTitle rappresenta la stringa di caratteri corrispondente al titolo dello schermo (NULL se non si desidera indicare nessun titolo); Gadgets contiene l'indirizzo della prima struttura Gadget in una lista che definisce tutti i gadget dello schermo. Infine, se si vuole utilizzare una propria bit map, occorre memorizzarne l'indirizzo nel parametro CustomBitMap (in caso contrario NULL).

Il parametro ViewMode, l'unico che non abbiamo ancora descritto, dev'essere inizializzato con un valore corrispondente alle caratteristiche dello schermo che si vuole aprire. In C questi valori sono rappresentati dalle comode definizioni (direttiva #define) presenti nei file INCLUDE, ma queste definizioni sono semplici valori numerici, che in Basic si possono esprimere senza problemi sia in notazione decimale sia in notazione esadecimale. La Tavola 1 riporta i valori che i parametri ViewMode e Type possono assumere, e i loro significati.

Si tenga presente che questi valori (in notazione binaria) rappresentano un particolare bit a 1. Pertanto, possono anche essere sommati per impostare a 1 più bit, cioè per attivare più flag. Per esempio, se si desidera che lo schermo sia in alta risoluzione orizzontale (640 pixel per linea) e in alta risoluzione verticale (512 pixel ottenuti in interlace), occorre memorizzare nel parametro ViewMode il valore -32764, ottenuto dalla somma algebrica di -32768 (HIRRES) e 4 (LACE); il valore -32768 corrisponde a una word nella quale solo il bit 15 è a 1, mentre il valore 4 corrisponde a una word nella quale solo il bit 2 è a 1: la loro somma

genera il valore -32764, che corrisponde appunto a una word nella quale sono impostati a 1 sia il bit 15 sia il bit 2.

Accesso alle librerie

Il sistema operativo dell'Amiga è strutturato in librerie "shared". Alcune, le più importanti, sono residenti su ROM, come Exec e Intuition, mentre altre sono residenti su disco e vengono caricate dal sistema operativo quando se ne presenta la necessità, senza che i programmi se ne rendano conto. Le librerie non residenti il sistema si aspetta di trovarle nella directory logica LIBS:, che normalmente è associata alla directory libs del disco sistema.

Vediamo i passi da compiere per accedere alle librerie shared da Basic.

A. Ogni libreria shared che s'intende impiegare da Basic dev'essere trasformata nell'opportuno file con estensione .BMAP richiesto dall'Amiga BASIC. Questa trasformazione viene eseguita utilizzando il tool ConvertFD e gli FDfiles presenti nel disco Extras. I file risultanti possono risiedere in qualsiasi directory del sistema. Si noti che la trasformazione vale per qualsiasi libreria shared, residente o non residente.

B. A volte si fa uso di funzioni che restituiscono un numero al programma chiamante (per esempio un codice che indica l'esito dell'operazione, oppure un indirizzo come nel caso della funzione OpenScreen). Se si desidera chiamare da Basic funzioni che restituiscono valori numerici, occorre dichiararle esplicitamente prima di aprire le librerie in cui sono contenute. Questo si fa con l'istruzione Basic: DECLARE FUNCTION fn() LIBRARY, dove fn() è il nome della funzione, eventualmente seguito dal tipo di valore restituito (in genere "&" per un long integer, cioè un valore intero da quattro byte). Tutte le funzioni che invece non restituiscono valori numerici non devono essere dichiarate.

C. Per aprire una libreria occorre indicare esplicitamente il cammino di ricerca (path) che individua la directory nella quale si trova il file .BMAP; quest'operazione può essere effettuata scrivendo il relativo cammino di ricerca prima del nome della libreria, oppure indicando come cammino quello corrente tramite l'istruzione Basic CHDIR.

L'istruzione Basic che permette di aprire una libreria shared è LIBRARY. Presupponendo che il cammino di ricerca sia già stato indicato tramite l'istruzione CHDIR, la sintassi dell'istruzione è LIBRARY "nome-libreria.library". Si noti che il nome della libreria dev'essere seguito obbligatoriamente dall'estensione .library (e non .BMAP come si potrebbe pensare). Eseguita questa istruzione, il programma può accedere alle funzioni contenute nella libreria. Si noti che per chiamare le funzioni che non restituiscono valori numerici si deve impiegare l'istruzione CALL, mentre per chiamare le altre è necessario comportarsi come se fossero funzioni Basic.

D. Nell'Amiga è sempre buona norma indicare al sistema quando non s'intende più utilizzare una sua risorsa. Quindi, anche da Basic è consigliabile chiudere tutte le librerie alla fine del programma, operazione che si può effettuare tramite l'istruzione in Basic LIBRARY CLOSE.

Apertura di un nuovo schermo

Per aprire un nuovo schermo occorre aprire la libreria intuition.library e servirsi della funzione OpenScreen in essa contenuta. Questa funzione si aspetta di ricevere come argomento l'indirizzo di una struttura NewScreen opportunamente inizializzata. In Basic, si può definire una struttura come un array monodimensionale, all'interno del quale i valori si dispongono secondo l'ordine imposto dalla struttura. Quando l'array è pronto, si ricava il suo indirizzo di memoria con un'opportuna istruzione Basic, e lo si passa a OpenScreen come argomento.

Occorre fare alcune precisazioni sull'organizzazione di questo array. I parametri contenuti nelle strutture standard dell'Amiga occupano quantità di byte differenti: possono richiedere un semplice byte, una word o una long word. Nella definizione dell'array occorre quindi fare in modo che ogni valore occupi il numero di byte previsto dal corrispondente parametro della struttura. Perché questo avvenga, inizializziamo l'array come array di interi, cioè un array nel quale ogni elemento occupa una word (due byte), e poi andiamo a memorizzarvi i dati rispettando l'organizzazione definita dalla struttura, tenuto però presente che in Basic un intero è costituito da due byte (i cui valori possibili vanno da -32768 a +32767).

Vediamo nei dettagli i passi da seguire per aprire uno schermo.

A. Aprire la libreria intuition.library rendendo

noto al sistema che la funzione OpenScreen restituisce un valore numerico (l'indirizzo della struttura Screen) al termine della sua esecuzione.

B. Dichiarare e riempire un array NewScreen% (contenente 15 elementi) con i dati previsti dalla struttura NewScreen.

C. Ricavarne il puntatore attraverso l'istruzione Basic VARPTR (NewScreen%).

D. Chiamare la funzione OpenScreen indicando come argomento l'indirizzo ottenuto con l'istruzione VARPTR. Trattandosi di una funzione che restituisce un valore numerico, occorre chiamarla con un'istruzione del tipo: $s\% = \text{OpenScreen}\& (\text{NewScreenPtr}\&)$, nella quale NewScreenPtr& è l'indirizzo di memoria a partire dal quale è memorizzato l'array NewScreen%, che rappresenta la struttura NewScreen.

Il valore che la funzione OpenScreen restituisce è l'indirizzo della struttura Screen che ha creato per gestire lo schermo. È necessario conservare questo indirizzo per poter usare altre funzioni di libreria riguardanti gli schermi. Se il valore restituito è zero, significa che il tentativo di apertura dello schermo è fallito (per esempio a causa della mancanza di memoria da destinare alle bit map); è quindi buona norma controllare questa condizione onde evitare spiacevoli inconvenienti.

Quando lo schermo non è più utile, è necessario ricordarsi di chiuderlo in modo tale da disimpegnare la memoria occupata. Per farlo, occorre chiamare la funzione CloseScreen con l'istruzione Basic: CALL CloseScreen% (s%), nella quale la variabile Basic s% deve contenere l'indirizzo della struttura Screen relativa allo schermo da chiudere.

Per aprire uno schermo, si può anche impiegare l'istruzione SCREEN dell'Amiga BASIC in alternativa alla chiamata diretta della funzione OpenScreen. L'unico problema è che diventa più complicato risalire all'indirizzo della struttura Screen relativa allo schermo. Vi si può riuscire aprendo sullo schermo una finestra Basic e ricavando l'indirizzo della relativa struttura Window tramite l'istruzione Basic $w\% = \text{WINDOW}(?)$. Infatti, in ogni struttura Window si trova un puntatore alla struttura Screen che definisce lo schermo sul quale la finestra è stata aperta. Questo puntatore inizia al byte 46 della struttura Window. L'istruzione Basic che ci consente di risalire all'indirizzo della struttura Screen di definizione dello schermo è: $s\% = \text{PEEK}(\text{WINDOW}(?) + 46)$. Rispetto al primo, questo secondo metodo è evidentemente sconsigliabile se la finestra che siamo costretti ad aprire non serve a nient'altro.

Infine, per effettuare certe operazioni sugli schermi è anche necessario conoscere gli indirizzi delle strutture RastPort e ViewPort associate al particolare schermo. Supponendo che l'indirizzo della struttura Screen sia stato memorizzato nella variabile s%, possiamo facilmente risalire agli indirizzi delle due strutture RastPort e ViewPort, entrambi contenuti all'interno della struttura Screen, con le seguenti istruzioni Basic:

```
rps% = s% + 84
vps% = s% + 44
```

L'esempio che segue apre uno schermo impiegando la funzione `OpenScreen`, e risale all'indirizzo della struttura `RastPort` per disegnare sullo schermo.

Esempio 1 Apertura di uno schermo (`OpenScreen`)

```
Librerie:
  DECLARE FUNCTION OpenScreen% LIBRARY
  LIBRARY "DF0:BMAPfiles/intuition.library"
  LIBRARY "DF0:BMAPfiles/graphics.library"

StrutturaNewScreen:
  DIM newScreen%(15)
  FOR i%=0 TO 15
    READ newScreen%(i%)
  NEXT i%
  DATA 0,25: 'angolo sup. sinistro schermo (LEFT, TOP)
  DATA 320,230: 'dimensioni schermo (WIDTH, HEIGHT)
  DATA 4,1: 'bit plane (DEPTH), colori (OR logico)
  DATA 0,271: 'modo (VIEW-MODE), custom screen (TYPE)
  DATA 0,0: 'caratteri (FONT)
  DATA 0,0: 'titolo dello schermo (DEFAULT-TITLE)
  DATA 0,0: 'struttura gadgets agganciata (GADGETS)
  DATA 0,0: 'struttura bit map agganciata (BITMAP)

AperturaNuovoSchermo:
  s%=OpenScreen% (VARPTR(newScreen%(0))): 'Puntatore
  allo schermo
  IF s%=0 THEN
    BEEP:PRINT "Apertura nuovo schermo fallita!"
    GOTO Fine
  END IF
  rps%=s%+84: 'Rast port schermo

EsempioUtilizzo:
  c%=1
  FOR i%=20 TO 200 STEP 12
    c%=c%+1:CALL SetAPen( rps%,c% )
    FOR j%=0 TO 320
      j% = INT( RND*1% )
      CALL Move( rps%,i%,0 )
      CALL Draw( rps%,i%,j% )
    NEXT j%
  NEXT i%

  PRINT "—Premi il tasto sinistro del mouse sulla
  finestra Output —"
  WHILE MOUSE(0)=0:WEND

Fine:
  CALL CloseScreen% (s%)
  LIBRARY CLOSE
END
```

Questo programma apre uno schermo tramite la funzione `OpenScreen` di `Intuition`, definendo una struttura (array) `NewScreen%` che contiene gli attributi dello schermo: risoluzione 320 x 230 (a partire dalla 25^a linea), con quattro bit plane (16 colori) e con `Type = 15 + 256 = 271`, cioè con gli attributi `CUSTOMSCREEN` e `SCREENQUIET` (senza barra titolo). Una volta finito il disegno dimostrativo nei vari colori, si può premere il tasto sinistro del mouse per uscire dal programma. Si noti che i file `.BMAP` vengono indicati come presenti all'interno di una directory `BMAPfiles` sul disco inserito nel disk drive `DF0:`. Questo cammino di ricerca dev'essere opportunamente modificato qualora i file `.BMAP` risiedano altrove.

Funzioni relative agli schermi

Vediamo ora una serie di funzioni contenute nella libreria `Intuition` che ci permettono di effettuare alcune interessanti operazioni sugli schermi.

Come abbiamo già spiegato, anche gli schermi possono essere sovrapposti come le finestre. Generalmente, gli schermi dispongono di due appositi gadget di profondità che consentono all'utente di cambiarne l'ordine di sovrapposizione. Queste operazioni, che di solito si svolgono con il mouse, possono essere svolte da un programma in Basic chiamando le funzioni `ScreenToBack(s%)` e `ScreenToFront(s%)` della libreria `Intuition` nel modo che abbiamo già visto. Queste funzioni fanno rispettivamente "indietreggiare" e "tornare in superficie" lo schermo del quale si è passato come argomento l'indirizzo della struttura `Screen`.

Esistono altre due funzioni che svolgono lo stesso ruolo, ma solo per lo schermo `Workbench`: `WBenchToFront()` e `WBenchToBack()`, le quali non richiedono alcun parametro dal momento che lo schermo `Workbench` è lo schermo di default. Queste due funzioni restituiscono il valore `NULL` (cioè zero) se lo schermo `Workbench` non era aperto all'atto della chiamata.

Un'altra operazione che si può svolgere dal Basic grazie a `Intuition` è lo spostamento verticale dello schermo. La funzione da impiegare è `MoveScreen(s%, delta-x%, delta-y%)`, nella quale `s%` deve contenere l'indirizzo della struttura `Screen` dello schermo, mentre `delta-x%` e `delta-y%` devono contenere rispettivamente la quantità di pixel orizzontali e verticali di cui si desidera muovere lo schermo.

Un'ultima coppia di funzioni molto interessanti sono: `result%=CloseWorkbench%` e `result%=OpenWorkbench%`, che ci consentono di chiudere e riaprire lo schermo `Workbench`, così da risparmiare all'occorrenza una quarantina di kilobyte di preziosa memoria chip, ovvero lo spazio di memoria occupato dallo schermo `Workbench`. Il valore `result%` restituito dalle funzioni (che dovranno quindi essere dichiarate come funzioni che restituiscono valori numerici) indica l'esito: se vale `NULL` significa che l'operazione non ha avuto successo. `CloseWorkbench` non ha mai successo se sullo schermo `Workbench` risulta presente anche una sola finestra. `OpenWorkbench` può non aver successo se il sistema non trova sufficiente memoria libera per riallocare lo schermo `Workbench`.

È bene precisare che queste due funzioni non hanno niente a che vedere con il programma `Workbench`, quello con il quale siamo abituati a interagire. In particolare, mandare in esecuzione la funzione `CloseWorkbench` non significa chiudere il programma `Workbench`, e chiamare la funzione `OpenWorkbench` non significa caricarlo da disco: durante l'uso di queste due funzioni l'esecuzione del `Workbench` non viene interrotta, come dimostra il prossimo programma.

Come abbiamo detto, la funzione `CloseWork-`

bench è un po' delicata in quanto può essere utilizzata solo a condizione che non sia aperta nessuna finestra nello schermo Workbench, e quindi neanche la finestra di output del programma. In ambiente Amiga BASIC questo si traduce nell'impossibilità di usare CloseWorkbench se non si provvede ad aprire per il Basic un nuovo schermo, e a chiudere la finestra List rimasta sullo schermo Workbench. Questo è il metodo che usa il programma d'esempio che segue.

Esempio 2

Chiusura dello schermo Workbench (CloseWorkbench)

```

Librerie:
DECLARE FUNCTION CloseWorkbench% LIBRARY
DECLARE FUNCTION OpenWorkbench% LIBRARY
LIBRARY "DF0:BMAPfiles/intuition.library"

AperturaSchermo:
SCREEN 1,320,200,2,1
WINDOW 1,,(0,20)-(279,150),15,1
s%=PEEK(L(WINDOW(7))+46)

CALL MoveScreen(s%,0,30)
PRINT "Trascina giu' questo nuovo schermo"
PRINT "e chiudi tutte le finestre presenti"
PRINT "sullo schermo Workbench.":PRINT

PRINT "Successivamente, riporta su questo"
PRINT "schermo e premi un tasto all'interno"
PRINT "di questa finestra."
WHILE INKEY$="" :WEND

PRINT:PRINT "Schermo Workbench aperto:"
PRINT "  memoria libera =": FRE(-1):PRINT
IF CloseWorkbench%=0 THEN
  PRINT "NON POSSO CHIUDERE LO SCHERMO WORKBENCH"
ELSE
  PRINT "Schermo Workbench chiuso:"
  PRINT "  memoria libera =": FRE(-1):PRINT
END IF

PRINT "Premi un tasto..."
WHILE INKEY$="" :WEND
Fine:
IF OpenWorkbench%=0 THEN
  PRINT "Non riesco a riaprire lo schermo Workbench"
END IF
WINDOW 1,,31,-1
SCREEN CLOSE 1
LIBRARY CLOSE
END

```

Viene aperto un nuovo schermo e poi una finestra per l'output dell'Amiga BASIC. Con questa operazione scompare dallo schermo Workbench la finestra di output, e appare sul nuovo schermo la barra menu dell'Amiga BASIC. A questo punto l'utente viene invitato a chiudere la finestra List ed eventualmente le altre finestre ancora presenti sullo schermo Workbench.

Ora il programma può chiamare con successo la funzione CloseWorkbench, che libera la memoria occupata dallo schermo Workbench. Effettuata quest'operazione, il programma mostra l'apprezzabile differenza di memoria libera fra la configurazione con schermo Workbench aperto e quella con schermo Workbench chiuso.

Infine, il programma chiama la funzione

OpenWorkbench per riaprire lo schermo Workbench, quindi chiude la finestra e lo schermo che aveva aperto, chiude la libreria ed esce. A questo punto, l'utente può rilanciare il programma e interagire con il Workbench come se niente fosse accaduto.

La struttura NewWindow

La struttura NewWindow è analoga alla struttura NewScreen: i programmi che desiderano aprire una finestra devono inizializzarla e passarla come argomento alla funzione OpenWindow, la quale restituisce l'indirizzo della struttura Window che ha creato per la gestione della finestra.

Al livello più basso, le finestre sono costituite da layer, cioè da bit map indipendenti le une dalle altre. Proprio per questa indipendenza, i layer possono essere spostati sullo schermo come fogli su un tavolo; e proprio come fogli possono essere sovrapposti fra loro. Fortunatamente, i programmatori possono disinteressarsi completamente dei layer, salvo in casi particolari, e gestire le finestre a un livello più alto.

Vediamo la definizione della struttura NewWindow riportata nel file INCLUDE intuition.h:

```

struct NewWindow
{
  SHORT LeftEdge, TopEdge;
  SHORT Width, Height;
  UBYTE DetailPen, BlockPen;
  ULONG IDCMPFlags;
  ULONG Flags;
  struct Gadget *FirstGadget;
  struct Image *CheckMark;
  UBYTE *Title;
  struct Screen *Screen;
  struct BitMap *BitMap;
  SHORT MinWidth, MinHeight;
  USHORT MaxWidth, MaxHeight;
  USHORT Type;
};

```

I primi sei parametri sono analoghi a quelli visti per la struttura NewScreen. I parametri IDCMPFlags e Flags contengono flag che descrivono la gestione della finestra e le sue caratteristiche. Se si desidera associare alla finestra una serie di gadget personalizzati, il puntatore FirstGadget dev'essere inizializzato con l'indirizzo di una lista di strutture Gadget. Title viene inizializzato con l'indirizzo della stringa di testo che costituisce il titolo della finestra. Screen è il puntatore che individua la struttura Screen che definisce lo schermo. BitMap dev'essere inizializzato con l'indirizzo di una struttura BitMap se si desidera impiegarne una particolare anziché quella di default. I quattro parametri che seguono definiscono le dimensioni minime e massime che la finestra può assumere quando viene ridimensionata. Infine, Type indica il tipo di schermo.

Nell'ambito di questo articolo avremo a che fare soltanto con le dimensioni, le caratteristiche, il titolo, il puntatore allo schermo, le dimensioni minime e massime e il tipo di schermo: sono

VALORI CHE PUÒ ASSUMERE IL PARAMETRO FLAGS

Definizione C	Esadec.	Decim.	Descrizione
WINDOWSIZE	0x0001	1	attiva il gadget di ridimensionamento
WINDOWDRAG	0x0002	2	attiva la barra di spostamento
WINDOWDEPTH	0x0004	4	attiva i gadget di profondità
WINDOWCLOSE	0x0008	8	attiva il gadget di chiusura finestra
BACKDROP	0x0100	256	finestra "incollata" allo sfondo
GIMMEZERO	0x0400	1024	origine delle coordinate sotto la barra titolo
BORDERLESS	0x0800	2048	finestra senza bordi
ACTIVATE	0x1000	4096	attiva la finestra dopo l'apertura
SMART_REFRESH	0x0000	0	ridisegna la finestra dopo che è stata "scoperta"
SIMPLE_REFRESH	0x0040	64	non ridisegna la finestra automaticamente

VALORI CHE PUÒ ASSUMERE IL PARAMETRO TYPE

Definizione C	Esadec.	Decim.	Descrizione
WBENCHSCREEN	0x0001	1	schermo Workbench
CUSTOMSCREEN	0x000F	15	schermo personalizzato

Tavola 2

parametri più che sufficienti per gestire le finestre con più flessibilità di quanto ci consenta l'Amiga BASIC, senza tuttavia complicare troppo le cose (situazione che renderebbe opportuno ricorrere al linguaggio C).

Nella Tavola 2 sono illustrate le costanti che possiamo impiegare per i parametri Flags e Type; come nel caso della struttura NewScreen, ognuna imposta un particolare bit, o flag, del parametro. Quelle che iniziano con la parola WINDOW attivano i vari gadget standard previsti da Intuition per agire sulla finestra (il gadget di ridimensionamento, la barra di spostamento, i gadget di profondità e il gadget di chiusura). BACKDROP serve a visualizzare la finestra "sul fondo", in modo che non possa coprire altre finestre. GIMMEZERO serve a fare in modo che l'origine delle coordinate all'interno della finestra corrisponda all'angolo in alto a sinistra dello spazio interno alla finestra, immediatamente sotto la barra di spostamento, come avviene con le finestre apribili dall'Amiga BASIC; se questo flag non viene impostato, l'origine delle coordinate diventa il punto più in alto a sinistra di tutta l'area occupata dalla finestra (si noti però che l'uso di GIMMEZERO occupa più memoria a parità di area). BORDERLESS serve a non far disegnare il bordo che racchiude la finestra. ACTIVATE fa in modo che la finestra risulti attiva non appena viene aperta, come avviene col comando WINDOW dell'Amiga BASIC. SMART_REFRESH e SIMPLE_REFRESH indicano al sistema cosa fare nel caso in cui la finestra venga sovrapposta da un'altra: la prima costante fa in modo che quando la finestra viene coperta sia lo stesso sistema operativo a occuparsi di salvare il contenuto della relativa bit map, allo scopo di rivisualizzarlo quando la parte coperta torna in superficie (si noti però che questo refresh automatico non avviene se

la finestra viene ridimensionata; in questo caso è il task che deve provvedere a ripristinare il contenuto della finestra, come avviene per la finestra CLI). La seconda costante, SIMPLE_REFRESH, specifica viceversa che l'operazione di refresh verrà effettuata dal programma (risparmiando così una quantità considerevole di memoria).

Le costanti relative al parametro Type indicano il tipo di schermo: WBENCHSCREEN o CUSTOMSCREEN. Indicando la prima costante, Intuition apre la finestra sullo schermo Workbench, qualunque sia il contenuto del parametro Screen; indicando invece la seconda, Intuition apre la finestra sullo schermo individuato dal parametro Screen.

Da Amiga BASIC è anche possibile utilizzare le funzioni della libreria Exec per entrare in attesa di messaggi. Questo comporta l'impiego di message port, messaggi, segnali... una complessa rete di comunicazione che permette al programma Basic di essere "avvisato" da Intuition per esempio quando l'utente seleziona un gadget della finestra. Realizzare un programma in Basic che sia in grado di elaborare messaggi provenienti da Intuition è però molto complicato, al punto che diventa più conveniente impiegare il linguaggio C. Quindi non ci soffermeremo su questo aspetto della programmazione in Amiga BASIC.

Vediamo ora com'è possibile aprire una finestra impiegando le funzioni di Intuition, cioè disponendo di un controllo molto più diretto di quanto consente l'istruzione WINDOW del Basic. Come per la gestione degli schermi, è sufficiente definire e inizializzare opportunamente un array che chiamiamo NewWindow%, ricavarne l'indirizzo memorizzandolo nella variabile NewWindow&, e indicarlo come argomento della funzione OpenWindow di Intuition. L'istruzione Basic è: `NewWindow% = OpenWindow(NewWindow%);` in w& si ottiene l'indirizzo della struttura Window che Intuition impiega per

gestire la finestra (si noti che se `w&` è uguale a zero significa che per qualche ragione la funzione non è riuscita ad aprire la finestra).

Se invece s'impiega l'istruzione `WINDOW` del Basic, si riesce ugualmente a risalire all'indirizzo della struttura `Window` di gestione della finestra tramite l'istruzione Basic: `w& = WINDOW(7)`. Grazie all'indirizzo `w&`, si può poi risalire all'indirizzo della struttura `RastPort` (contenuto nella struttura `Window`) tramite l'istruzione: `rp& = PEEKL(w&+50)`.

Si noti che se la finestra è stata aperta con l'istruzione Basic `WINDOW`, l'indirizzo della struttura `Window` lo possiamo ricavare anche con l'istruzione: `rp& = WINDOW(8)`.

Il programma che ora illustriamo apre una finestra sullo sfondo impostando i flag `BACKDROP` e `BORDERLESS` per ottenere un interessante effetto.

Esempio 3

Apertura di una finestra (OpenWindow)

```

Librerie:
DECLARE FUNCTION OpenWindow& LIBRARY
LIBRARY "DF0:BMAPfiles/intuition.library"
LIBRARY "DF0:BMAPfiles/graphics.library"

StrutturaNewWindow:
DIM newWindow%(23)
FOR i%=0 TO 23
  READ newWindow%(i%)
NEXT i%
DATA 0, 0: ' SHORT dimensioni
DATA 640, 256: ' SHORT
DATA 1: ' UBYTE (UBYTE#UBYTE=SHORT)
DATA 0, 0: ' ULONG
DATA 0, 6400: ' ULONG FLAGS=BACKDROP|BORDERLESS
DATA 0, 0: ' ULONG
DATA 0, 0: ' ULONG
DATA 0, 0: ' ULONG
DATA 0, 0: ' ULONG
DATA 0, 0: ' ULONG
DATA 20, 20: ' SHORT dimensioni minime
DATA 640, 256: ' USHORT dimensioni massime
DATA 1: ' USHORT NBENCHSCREEN

AperturaNuovaFinestra:
MyW% = OpenWindow&( VARPTR(newWindow%(0)) )
IF MyW%=0 THEN
  BEEP:PRINT "Apertura nuova finestra fallita!"
  GOTO Fine
END IF
rp% = PEEKL( MyW% + 50 )

EsempioUtilizzo:
FOR j%=0 TO 640 STEP 10
  j% = j% + 2.5
  CALL Move( rp%, 1%, 0)
  CALL Draw( rp%, 0, j%)
NEXT j%
FOR j%=0 TO 256 STEP 4
  j% = j% + 2.5
  CALL Move( rp%, 1%, 256)
  CALL Draw( rp%, 640, j%)
NEXT j%

PRINT "--Premi il tasto sinistro del mouse sulla
finestra Output--"
WHILE MOUSE(0)=0 WEND
Fine
CALL CloseWindows MyW%
LIBRARY CLOSE
END

```

Questo programma costruisce un array `NewWindow%` contenente 23 elementi e chiama la funzione `OpenWindow` per designare una finestra di tipo `BACKDROP`, che risiede cioè sempre e solo sullo sfondo; su di essa viene poi disegnato un retino che dà l'impressione di lavorare direttamente sullo schermo `Workbench`.

Funzioni relative alla gestione delle finestre

La prima funzione che analizziamo permette di cambiare il nome della finestra e/o del relativo schermo senza influire sul contenuto della finestra (operazione che in Basic è impossibile in quanto il comando `WINDOW` ridisegna ogni volta il contenuto della finestra). La sintassi della funzione è la seguente: `SetWindowTitles(w&, window_name_ptr&, screen_name_ptr&)`, dove `w&` è il puntatore alla struttura `Window` relativa alla finestra sulla quale desideriamo agire; `window_name_ptr&` e `screen_name_ptr&` sono gli indirizzi rispettivamente del nome della finestra e del nome dello schermo. Occorre notare che non è possibile passare direttamente una stringa alle funzioni di libreria, ma si può aggirare l'ostacolo specificando il puntatore alla locazione di memoria in cui è contenuto il primo carattere della stringa. La fine dell'insieme di caratteri ASCII è segnalata dal terminatore `CHR$(0)`. Per ricavare l'indirizzo relativo all'inizio di una stringa, l'Amiga BASIC dispone della funzione `SADD(stringa$)`. Se uno dei due puntatori viene inizializzato con il valore `-1`, il corrispondente titolo viene lasciato inalterato.

Un'altra operazione che Intuition ci consente di compiere facilmente è la modifica dei limiti di ridimensionamento di una finestra senza che occorra ridefinirla. La funzione di cui dobbiamo servirci è `WindowLimits`, che prevede la seguente sintassi: `WindowLimits(w&, min-width%, min-height%, max-width%, max-height%)`. Oltre al già citato `w&`, gli altri valori si riferiscono alle dimensioni minime e massime (in larghezza e altezza) che la finestra può assumere quando l'utente la ridimensiona con il mouse. Questa funzione restituisce il valore `NULL` qualora l'operazione non abbia esito positivo, per esempio se i limiti minimi indicati come argomenti sono maggiori dell'attuale dimensione della finestra, oppure quelli massimi sono minori.

Bisogna fare attenzione a usare questa funzione sulla finestra `Output` del Basic, perché se imponiamo una misura verticale superiore alle 200 righe NTSC, una volta che la finestra viene chiusa ogni tentativo di riaprirla darà inevitabilmente l'errore "Illegal function call". A questa stessa situazione si perviene se per esempio si estende manualmente la finestra `List` dell'Amiga BASIC: dopo che l'avremo chiusa non riusciremo a riaprirla se non selezionando l'opzione `Start` del menu `Run`, oppure l'opzione `New` del menu `Project`.

Le due funzioni di Intuition `WindowToFront(w&)` e `WindowToBack(w&)` sono analoghe a quelle viste per gli schermi. Il loro scopo è portare la finestra `w&` in

superficie, o spostarla al di sotto delle altre.

La funzione `MoveWindow(w%, delta-x%, delta-y%)` serve a spostare la finestra in un'altra posizione dello schermo. I parametri `delta-x%` e `delta-y%` indicano lo spostamento della finestra lungo la coordinata delle ascisse e delle ordinate.

È inoltre possibile modificare le dimensioni della finestra tramite la funzione `SizeWindow(w%, delta-x%, delta-y%)`, in cui - è ormai ovvio - i due parametri "delta" determinano il numero di pixel di cui dev'essere ingrandita o rimpicciolita (se il valore è negativo) la finestra individuata dall'indirizzo `w%`.

Per concludere citiamo la funzione `ActivateWindow(w%)`, che serve a rendere attiva la finestra indicata da `w%`; ovviamente esiste anche una `CloseWindow(w%)`, che ci consente di chiuderla.

Ridirezione dell'output

Fin qui si è visto come utilizzare le funzioni di libreria per gli schermi e per le finestre semplicemente passando come parametro l'indirizzo della struttura appropriata. Per usufruire di talune istruzioni Basic che producono un output (come da esempio, FILES), sarebbe utile poter "ingannare" l'Amiga BASIC definendo come finestra Output una qualsiasi altra finestra o schermo. In questo modo gli effetti di tutti i comandi Basic verrebbero dirottati sulla finestra aperta dal nostro programma anziché comparire sulla finestra Output dell'Amiga BASIC.

Ricordando che con l'istruzione Basic `WINDOW(8)` otteniamo l'indirizzo della struttura `RastPort` relativa alla finestra Output, possiamo ricavare l'indirizzo del parametro `BitMap` in essa contenuto tramite l'istruzione: `BM1oc% = WINDOW(8) + 4`. Successivamente, possiamo ricavare il contenuto di tale parametro, cioè l'indirizzo della struttura `BitMap`, tramite l'istruzione `oldBM% = PEEKL(BM1oc%)`. Questo indirizzo dev'essere memorizzato altrove per poterlo ricollocare successivamente nel parametro `BitMap`. Ora passiamo a modificarlo in modo che l'output dei comandi Amiga BASIC venga diretto su un altro schermo. Se l'indirizzo della struttura Screen relativa a questo schermo è contenuto nella variabile `s%`, la ridirezione dell'output si ottiene con la seguente istruzione: `POKEL(BM1oc%), PEEKL(s%+88)`.

L'indirizzo della struttura Screen lo otteniamo direttamente come valore restituito dalla funzione `OpenScreen()`; se invece per aprire lo schermo abbiamo impiegato l'istruzione `SCREEN`, occorre aprirvi sopra una finestra che ci consenta di ricavare l'indirizzo della struttura Screen con l'istruzione: `s% = PEEKL(WINDOW(7)+46)`. Ottenuto questo indirizzo, la finestra può essere chiusa.

Un ulteriore metodo consiste nell'impiegare l'indirizzo base della libreria Intuition. Questo metodo però può essere impiegato soltanto quando si ha la certezza che lo schermo sul quale vogliamo ridirigere l'output è l'unico oltre allo schermo Workbench di default. Dato che l'indirizzo

base della libreria Intuition è contenuto nella variabile `IntuitionBase&`, l'istruzione Basic per ottenere l'indirizzo della struttura Screen relativa al nuovo schermo è la seguente: `s% = PEEKL(PEEKL(IntuitionBase+56))`. L'indirizzo base della libreria Intuition si può facilmente ricavare impiegando la funzione `OpenLibrary` della libreria Exec (per la quale bisognerà crearsi l'apposito file `.BMAP`). Questa funzione ha la seguente sintassi: `IntuitionBase = OpenLibrary(library-name-ptr%, version%)`, dove `library-name-ptr%` è il puntatore alla stringa contenente il nome della libreria da aprire, mentre `version%` in genere è impostato a zero. Da Basic, l'istruzione che dobbiamo impiegare è la seguente: `IntuitionBase% = OpenLibrary$(SADD("intuition.library"+CHR$(0)),0)`.

Infine facciamo presente che l'output, sebbene sia ridiretto verso un altro schermo, viene ugualmente influenzato dalle operazioni che compiamo con il mouse sulla vera finestra Output dell'Amiga BASIC.

Per concludere, offriamo una piccola curiosità senza alcun commento: provate a scrivere qualche valore nella locazione corrispondente all'indirizzo `(IntuitionBase%+76)` e date un RUN senza nessun programma.

Esempio 4 Ridirezione dell'output

```
Librerie:
LIBRARY "DF0:BMAPfiles/graphics.library"
LIBRARY "DF0:BMAPfiles/intuition.library"

Preparativi:
WINDOW CLOSE 1
WINDOW 1,,(0,0)-(617,185),31,-1
SCREEN 1,320,200,2,1
WINDOW 4,,(0,12)-(310,23),0,1:s%=PEEKL(WINDOW(7)
+46)
CALL MoveScreen(s%, 0, 50)
PRINT "Scrittura ridiretta"
PRINT "Clicka sulla finestra Output"
FOR r=1 TO 6000:NEXT r
WINDOW CLOSE 4

RidirezioneOutput:
WHILE MOUSE(0)=0:WEND
BM1oc% = WINDOW(8)+4
oldBM% = PEEKL(BM1oc%)
POKEL(BM1oc%),PEEKL(s%+88)
FILES:PRINT "Questo sarebbe dovuto andare a Output"
POKEL(BM1oc%),oldBM%

Fine:
FOR r=0 TO 6000:NEXT r
SCREEN CLOSE 1
LIBRARY CLOSE
END
```

Questo programma apre un nuovo schermo e su questo una finestra con le istruzioni dell'Amiga BASIC (`SCREEN` e `WINDOW`). La finestra viene usata soltanto per ricavare l'indirizzo della struttura Screen relativa allo schermo, e viene subito chiusa. Successivamente l'output del comando FILES viene provvisoriamente ridiretto su questo schermo anziché sulla finestra Output dell'Amiga BASIC.

LISTATI: MANDELBROT C-64/C-128

FRATTALI IN INTERLACE CON 320 X 200 PIXEL A QUATTRO COLORI

L'insieme a dimensione frazionale più famoso del mondo in due versioni per il C-64 e il C-128, che offrono l'alta risoluzione e un metodo software per superare il limite dei due colori

di Dario Greggio

I frattali sono quei bellissimi oggetti con mille sfumature che si vedono un po' dappertutto sui monitor dei computer. Il loro successo può essere spiegato molto facilmente: in una formula lunga meno di dieci caratteri riassumono un'elevata complessità matematica, un aspetto grafico sorprendente e una vera e propria sfida alle capacità di ogni computer. In altre parole hanno tutte le carte in regola per rientrare nella categoria delle cose belle e complesse ma non troppo complicate... un po' come il gioco degli scacchi.

Per quel che riguarda il C-64/128, dinanzi alle apparentemente limitate potenzialità dell'hardware, quasi nessuno ha mai cercato di ottenere effetti frattali sui computer a 8 bit della CBM. Chi scrive ha deciso invece di accettare la sfida. È vero che non esiste un compilatore C serio per questi computer, e il C sarebbe l'unico linguaggio d'alto livello adatto a garantire una velocità sufficiente, ma è vero anche che la potenza di elaborazione in linguaggio macchina di una CPU 6502 funzionante a 1 MHz eguaglia quella di un 68000 dotato di un clock pari a un terzo di quello installato sull'Amiga. Abbiamo quindi cominciato a sospettare che un programma interamente scritto in LM avrebbe dato risultati degni di nota sia sul C-64 sia sul C-128. Rimaneva però il problema della grafica: pochi colori nell'alta risoluzione standard, pochi pixel in multicolor.

La risoluzione super alta

Il problema, in effetti, non era secondario: vedere i frattali in bianco e nero ci avrebbe riportato ai pionieristici anni Sessanta, quando il signor Benoit Mandelbrot teorizzò per la prima volta, senza poterli vedere, i suoi oggetti a

dimensione frazionale. Il modo multicolor, invece, avrebbe fornito i quattro colori indispensabili ma avrebbe ridotto i preziosi filamenti del disegno a rozze e sgradevoli scalette di pixel.

Osservando l'Amiga, abbiamo pensato che sarebbe stata una bella sfida tentare di utilizzare un trucco analogo all'interlace anche con il VIC, il chip video del C-64 e del C-128 nel modo a 40 colonne. Sapevamo che non era possibile visualizzare alternativamente due schermate sfalsate di mezzo pixel come si fa sull'Amiga, però a noi bastava l'alta risoluzione standard di 320 x 200 pixel... purché aumentassero i colori.

L'idea è piuttosto semplice: alternando ad alta velocità due schermate da 320 x 200 pixel, ognuna dotata di due colori, si ottiene ancora una schermata da 320 x 200 pixel nella quale appaiono quattro colori, formati dalla sovrapposizione delle due coppie di colori che compongono le due schermate. In pratica, si riescono a ottenere quattro colori per sintesi additiva.

Per alternare due schermate alla massima velocità, occorre sincronizzarsi con il momento durante il quale il pennello elettronico non scandisce alcuna linea di schermo (perché altrimenti in alcuni istanti lo schermo apparirebbe suddiviso in due parti, una occupata dalla prima schermata e una occupata dalla seconda). Abbiamo quindi creato la solita routine in LM per la gestione degli interrupt del raster video. Si tratta di una routine che a ogni scansione verticale dello schermo cambia il blocco di memoria visibile dal chip video, in modo che alla scansione successiva l'immagine che si forma sullo schermo non sia quella appena visualizzata. Come risultato si ottiene uno schermo sfarfallante a 25 Hz che offre due immagini sovrapposte da 320 x 200 pixel, ognuna con due

colori propri che sovrapponendosi generano ai nostri occhi quattro colori. La cosa più interessante di questa particolare schermata grafica è la generazione dei colori che, essendo ottenuti per combinazione e sovrapposizione dei 16 colori originali del C-64, sono del tutto nuovi e non sarebbero generabili affidandosi semplicemente all'hardware. Lo sfarfallio purtroppo è consistente, anche se un'opportuna scelta di colori, attenua l'effetto. I due programmi che accompagnano l'articolo, uno per il C-64 e uno per il C-128, sfruttano questo singolare metodo per visualizzare l'insieme di Mandelbrot.

L'insieme di Mandelbrot in LM

Una volta escogitato il sistema per disporre dei colori e della risoluzione necessaria, la stesura in linguaggio macchina del programma di calcolo del frattale di Mandelbrot è diventata una sfida molto più stimolante. Lo scoglio principale che rimaneva da superare per realizzare l'algoritmo di calcolo della regione frattale era la scrittura di opportune routine di calcolo in virgola mobile o, al limite, anche in virgola fissa ma che comunque non fossero limitate agli interi. La formula che permette di calcolare gli elementi dell'insieme di Mandelbrot è: $z = z^2 + c$, dove z e c sono numeri complessi. Come si può notare, compaiono nella formula operazioni di somma, sottrazione, moltiplicazione ed elevamento a potenza. A livello di codici macchina, però, la CPU del C-64 non consente di eseguire altro che somme e sottrazioni d'interi, e quindi occorre una libreria di funzioni matematiche che svolgano anche operazioni riguardanti numeri decimali.

Tutte queste operazioni vengono normalmente svolte dall'interprete Basic, e quindi nelle ROM di sistema del C-64 dovevano esistere le routine necessarie. Avendo a disposizione il disassemblato del Kernel, è stato abbastanza facile rintracciare gli indirizzi di chiamata e i parametri necessari per l'impiego di queste routine. Le routine utilizzate eseguono calcoli su numeri in virgola mobile. Infatti, poiché il sistema operativo del C-64 contiene le routine di base per i calcoli in virgola mobile, abbiamo evitato di scrivere nuove routine per i calcoli in virgola fissa. Questo però avrebbe portato qualche vantaggio in termini di velocità, dal momento che il calcolo dei frattali richiede al massimo una cifra prima della virgola.

Nel programma sono stati presi alcuni accorgimenti per migliorare la velocità di esecuzione. Per esempio, le pagine grafiche vengono cancellate durante la fase d'inizializzazione, il che permette, durante il tracciamento dell'immagine frattale, di eseguire solo l'operazione di accensione dei pixel e di evitare quella di spegnimento. Il programma inoltre calcola l'indirizzo del byte contenente il generico pixel della pagina video solo ogni otto pixel, dal momento che gli otto pixel si trovano all'interno dello stesso byte a causa della particola-

re gestione della memoria video del C-64. Sono infine stati disabilitati gli interrupt, sempre allo scopo di aumentare la velocità di esecuzione.

Si sono poi adottati diversi "trucchi" suggeriti da un esame più approfondito della formula di Mandelbrot, tutti allo scopo di evitare calcoli superflui. Prima di descrivere queste scorciatoie software, riepiloghiamo brevemente la strada da seguire per calcolare un insieme di Mandelbrot.

Si esplora una zona del piano complesso, cioè il piano cartesiano che ha sull'asse delle ascisse i numeri reali e sull'asse delle ordinate i numeri immaginari. Per ogni punto (la distanza tra un punto e il successivo dipende dalla precisione che vogliamo ottenere) si calcola la successione esaminata poco sopra: z (numero complesso, inizialmente uguale a $0 + 0i$) = $z^2 + c$ (numero complesso corrispondente alle coordinate del punto sul piano). Possono verificarsi due casi: la successione diverge dopo poche iterazioni (in tal caso si memorizza il numero delle iterazioni effettuate), oppure la successione non diverge dopo un numero stabilito di cicli (allora si assume che non divergerà mai e si utilizza un valore convenzionale per indicare questa situazione).

Va detto che i punti appartenenti all'insieme di Mandelbrot, secondo la definizione matematica, sono quelli in cui la successione non diverge mai. Tuttavia, i punti più spettacolari da visualizzare sono quelli in cui la successione diverge dopo un numero finito d'iterazioni, e quindi nel nostro programma visualizziamo anche questi ultimi.

Tutti i programmi di calcolo dell'insieme esplorano il piano complesso (con un'accuratezza maggiore o minore a seconda della risoluzione) applicando a ogni pixel la routine di calcolo della successione. Questa rappresenta il ciclo più interno, quello che si deve ottimizzare in termini di tempi d'esecuzione. Considerando di memorizzare le coordinate del particolare punto del piano (corrispondente a un pixel sullo schermo) nella variabile complessa c , la routine di calcolo si può rappresentare simbolicamente nel modo seguente:

```
WHILE |z| < 2 AND Conto < 100
|
  z = z^2 + c
  Conto = Conto + 1
|
```

Il ciclo viene ripetuto finché il modulo del numero complesso rimane minore di 2 oppure fino a quando non si supera il valore massimo d'iterazioni consentite, qui posto pari a 100, che rappresenta la precisione con la quale desideriamo venga effettuato il calcolo. Poiché il calcolo del modulo di un numero complesso implica una radice quadrata e la somma di due quadrati, si ha a che fare con un'operazione piuttosto lunga a livello di cicli macchina. È tuttavia possibile evitare di calcolare la radice quadrata se si confronta il quadrato del modulo del numero complesso z con 4.

Facendo uso di variabili temporanee, si possono

poi risparmiare altre operazioni utilizzando i due quadrati calcolati durante l'operazione di elevamento a potenza anche per il calcolo del modulo. Infine, ma questo è proprio un artificio derivante dalla rappresentazione dei numeri in virgola mobile mantenuta internamente dal computer, è possibile moltiplicare per due un numero in virgola mobile semplicemente incrementando di uno il byte che ne rappresenta l'esponente.

Il programma da noi realizzato impiega circa cinque ore per esplorare tutta la regione compresa tra le coordinate $(-2,-2)$ e $(2,2)$, ma questo tempo si riduce di circa il 50 per cento se il programma viene eseguito su un C-128 in modo fast. Si tratta di prestazioni paragonabili a quelle ottenute in Pascal su un IBM XT funzionante a 8 MHz.

Le opzioni di contorno

Il programma *Mandelbrot*, nelle due versioni per C-64 e C-128, contiene le due routine in LM ora descritte e alcune altre cosette che servono per la gestione delle immagini.

Abbiamo infatti a disposizione un menu con cinque scelte. Le prime due servono a caricare e memorizzare immagini grafiche anche non generate con questo programma. Il formato è analogo a quello adottato dal pacchetto *Doodle II*, ossia si tratta di file lunghi 9K, composti da un kilobyte riguardante i colori e da altri 8 per la pagina bit map. Naturalmente occorrono due di questi file per ogni schermata (al cui nome il programma antepone i prefissi "FF1" e "FF2"). Nella versione per il C-64, in mancanza d'istruzioni in Basic per il caricamento e la registrazione di blocchi di memoria, abbiamo utilizzato una routine che byte dopo byte esegue il trasferimento da file a memoria e viceversa. Sempre con la versione per il C-64, abbiamo scartato alcune aree di memoria che non potevano essere utilizzate per i dati grafici; infatti quando il chip del video accede a quelle zone, in realtà accede alla ROM caratteri. La terza opzione esegue il calcolo della regione frattale: dapprima copia i dati della routine (la quale non è "rientrante", cioè dev'essere reinizializzata ogniqualevolta la si usa), poi la manda in esecuzione. Alla fine attiva la sezione che genera la visualizzazione nella nostra risoluzione speciale e ci permette di variane i colori, cioè attiva automaticamente l'opzione successiva. L'ultima opzione del menu ci permette di visualizzare una schermata a quattro colori, e di alterarne i colori attraverso l'uso dei tasti da 1 a 4. Ogni pressione di questi tasti causa l'incremento del codice di colore relativo nella pagina in alta risoluzione (non ha ovviamente alcun senso chiamarli colori di sfondo e di primo piano). Quando desideriamo tornare al menu, è sufficiente premere la barra spaziatrice.

Vediamo ora come si può agire sul programma di tracciamento per variane alcuni parametri. Il numero 100, che compare come quarto numero nelle istruzioni DATA, rappresenta il massimo

numero d'iterazioni effettuate durante il calcolo della successione. Riducendo tale valore si aumenta la velocità di esecuzione, anche se si perde in precisione e accuratezza dei particolari, mentre aumentandolo (il limite è 255) crescono linearmente il tempo d'esecuzione e anche la precisione: 100 è un compromesso per ottenere un buon risultato in un tempo ragionevole.

I venti numeri che seguono sono la rappresentazione in virgola mobile dei quattro estremi della zona di piano complesso che viene esaminata dalla routine. Nell'ordine sono: ascissa minima, ascissa massima, ordinata minima e ordinata massima. I valori impostati sono quelli tipici dei calcoli relativi all'insieme di Mandelbrot, rispettivamente -2 , 2 , -2 e 2 . Per alterarli occorre avere un po' di dimestichezza con la memoria del C-64/128, e tenere conto del checksum che viene eseguito.

In pratica, quando dobbiamo stabilire quale cinquina di byte rappresenta, per esempio, il valore decimale $-1,7$, il metodo più semplice consiste nel lasciar fare la conversione al sistema, e leggere poi i byte che ci interessano. Dapprima si legge qual è la locazione di memoria a partire dalla quale verrà memorizzata la prossima variabile numerica: si ottiene scrivendo PRINT PEEK(46)*256+PEEK(45). Successivamente si assegna a una variabile di comodo il valore desiderato, $GO = -1.7$. A questo punto, a partire dall'indirizzo ricavato in precedenza troveremo i primi due caratteri costituenti il nome della variabile e in seguito i cinque byte della rappresentazione in virgola mobile di $-1,7$: sarà sufficiente leggerli e riportarne i valori nel listato. È possibile ampliare il listato per leggere in input le due coppie di coordinate, effettuare la conversione e inserire a livello di codice oggetto i valori richiesti (si tenga presente che nel caso del C-64 il programma Basic non può crescere oltre i 24 blocchi su disco, mentre con il C-128 non può crescere oltre i 29 blocchi). Gli utenti del C-128 devono ricordarsi di lavorare nel banco di memoria delle variabili, ossia nel banco 1. Possono inoltre abbreviare la procedura sopra descritta assegnando, senza altri interventi, il valore desiderato a una variabile di comodo e leggendone poi l'indirizzo con la funzione POINTER(<nome variabile>): il nostro numero in virgola mobile si trova all'indirizzo restituito da questa funzione. I tre numeri che seguono, nelle istruzioni DATA, rappresentano la larghezza dello schermo (due byte) e l'altezza (un byte), qui fissate a 320 pixel e a 200. Se si desidera modificarli, bisogna fare attenzione a non aumentarli oltre questi valori.

Ulteriori sviluppi

Le possibilità aperte dal nuovo modo grafico che abbiamo illustrato sono notevoli. Tra le altre cose, si possono ottenere sullo schermo anche i colori prodotti dall'hardware del C-64, accanto a quelli ottenuti per sovrapposizione: basta che ai pixel ai

quali si vuole assegnare uno dei 16 colori "hardware" corrisponda nelle due schermate lo stesso colore, che sovrapponendosi a se stesso non genera nuove tinte. Spingendosi oltre, si potrebbero realizzare programmi in grado di disegnare un'immagine da 320 x 200 pixel con quattro colori scelti da una palette di 136 (16 hardware più 120 per sovrapposizione). Si può anche pensare di utilizzare questo sistema per i giochi che non presentano animazioni, come adventure o simili. Infine, anche il modo d'impiegare le routine matematiche del Kernel può costituire un utile spunto per i programmatori che hanno bisogno di effettuare nelle loro applicazioni calcoli con numeri decimali.

Listato 1: Mandelbrot 64

```

10 rem : Mandelbrot in interlace
20 rem : * versione C-64 *
32 poke53265,27:poke53272,21
35 restore:c=0:for i=49152 to 50130:reada:
c=c+a:pokei,a:next
37 if c<>112240 then print"errore nei data...":
end
50 printchr$(147)"1. Carica immagine"
60 print"2. Salva immagine"
70 print"3. Calcola immagine"
80 print"4. Visualizza immagine"
85 print"5. Fine"
90 print:print"Scegli:"
100 geta$:ifa$<"1"ora$>"5"then100
110 onval(a$)gosub200,300,400,500,600
120 run
200 print:input"Nome del file ":n$
210 open2,8,0,"ff1"+left$(n$,13)+",p,r"
211 get#2,a$:get#2,a$
212 for i=1024 to 2047:get#2,a$:
pokei,asc(a$+chr$(0)):next
214 for i=8192 to 16383:get#2,a$:
pokei,asc(a$+chr$(0)):next
215 close2
216 open2,8,0,"ff2"+left$(n$,13)+",p,r"
218 for i=17408 to 18431:get#2,a$:
pokei,asc(a$+chr$(0)):next
220 for i=24576 to 32767:get#2,a$:
pokei,asc(a$+chr$(0)):next
225 close2
230 goto500
300 print:input"Nome del file ":n$
310 open2,8,1,"ff1"+n$+",p,w"
311 print#2,chr$(0)chr$(28):
312 for i=1024 to 2047:print#2,chr$(peek(i)):
next
314 for i=8192 to 16383:print#2,chr$(peek(i)):
next
315 close2
316 open2,8,1,"ff2"+n$+",p,w"
317 print#2,chr$(0)chr$(92):
318 for i=17408 to 18431:print#2,chr$(peek(i)):
next
320 for i=24576 to 32767:print#2,chr$(peek(i)):
next
325 close2
330 return
400 restore:c=0:for i=49152 to 50130:reada:
c=c+a:pokei,a:next
410 if c<>112240 thenprint"errore nei data...":
end

```

```

420 poke53265,peek(53265)or32
430 poke53272,24
440 sys49152
500 poke53265,peek(53265)or32
510 poke53272,24
520 sys49920
530 return
600 end
1000 data76,162,192,100,130,128,0,0,0,130,0,0
1010 data0,0,130,128,0,0,0,130,0,0,0,0,64,1,200
1020 data0,131,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1030 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1040 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1050 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1060 data54,87,128,64,32,16,8,4,2,1,0,32,64,33
1070 data128,34,192,35,0,37,64,38,128,39,192,40
1080 data0,42,64,43,128,44,192,45,0,47,64,48,128
1090 data49,192,50,0,52,64,53,128,54,192,55,0
1100 data57,64,58,128,59,192,60,0,62,169,32,133
1110 data252,169,96,133,254,160,0,132,251,132
1120 data253,152,145,251,145,253,200,208,249,230
1130 data252,230,254,165,252,201,64,208,238,169
1140 data4,133,252,169,68,133,254,173,102,192
1150 data145,251,173,103,192,145,253,200,208,243
1160 data230,252,230,254,165,252,201,8,208,233
1170 data120,169,9,160,192,32,162,187,169,4,160
1180 data192,32,80,184,32,88,188,162,63,160,192
1190 data32,212,187,169,19,160,192,32,162,187
1200 data169,14,160,192,32,80,184,32,88,188,162
1210 data68,160,192,32,212,187,172,24,192,173
1220 data25,192,32,145,179,169,63,160,192,32,15
1230 data187,162,73,160,192,32,212,187,172,26
1240 data192,173,27,192,32,145,179,169,68,160
1250 data192,32,15,187,162,78,160,192,32,212,187
1260 data169,4,160,192,32,162,187,162,38,160,192
1270 data32,212,187,169,4,160,192,32,162,187,162
1280 data38,160,192,32,212,187,169,14,160,192
1290 data32,162,187,162,48,160,192,32,212,187
1300 data169,38,160,192,32,162,187,162,43,160
1310 data192,32,212,187,169,33,160,192,32,162
1320 data187,162,58,160,192,32,212,187,162,53
1330 data160,192,32,212,187,169,0,133,250,169
1340 data53,160,192,32,162,187,169,53,160,192
1350 data32,40,186,162,83,160,192,32,212,187,162
1360 data88,160,192,32,212,187,169,58,160,192
1370 data32,162,187,169,58,160,192,32,40,186,162
1380 data93,160,192,32,212,187,169,83,160,192
1390 data32,80,184,169,43,160,192,32,103,184,162
1400 data83,160,192,32,212,187,169,58,160,192
1410 data32,162,187,169,53,160,192,32,40,186,230
1420 data97,169,48,160,192,32,103,184,162,58,160
1430 data192,32,212,187,169,83,160,192,32,162
1440 data187,162,53,160,192,32,212,187,230,250
1450 data165,250,205,3,192,240,26,169,88,160,192
1460 data32,162,187,169,93,160,192,32,103,184
1470 data169,28,160,192,32,91,188,16,3,76,145
1480 data192,173,98,192,41,7,170,208,49,173,100
1490 data192,74,74,41,254,168,185,112,192,133
1500 data251,185,113,192,133,252,173,100,192,41
1510 data7,24,101,251,133,251,144,2,230,252,24
1520 data173,98,192,41,248,101,251,133,251,165
1530 data252,109,99,192,133,252,165,250,41,1,240
1540 data9,189,104,192,160,0,17,251,145,251,165
1550 data250,41,2,240,21,165,252,73,64,133,252
1560 data189,104,192,160,0,17,251,145,251,165
1570 data252,73,64,133,252,238,98,192,208,3,238
1580 data99,192,169,38,160,192,32,162,187,169
1590 data73,160,192,32,103,184,162,38,160,192
1600 data32,212,187,173,99,192,205,25,192,240
1610 data3,76,106,193,173,98,192,205,24,192,240
1620 data7,76,106,193,169,0,141,99,193,141,99
1630 data192,238,100,192,169,14,160,192,32,162
1640 data187,169,78,160,192,32,103,184,162,14

```

```

1650 data160,192,32,212,187,173,100,192,205,26
1660 data192,240,3,76,78,193,88,96,0,0,0,0,0
1670 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1680 data169,192,141,20,3,169,195,141,21,3,169
1690 data0,141,18,208,173,17,208,41,127,141,17
1700 data208,169,127,141,13,220,169,1,141,26,208
1710 data88,165,197,170,201,56,208,11,230,251
1720 data165,251,41,15,133,251,24,144,48,201,59
1730 data208,11,230,252,165,252,41,15,133,252
1740 data24,144,33,201,8,208,11,230,253,165,253
1750 data41,15,133,253,24,144,18,201,11,240,6
1760 data201,60,240,65,208,198,230,254,165,254
1770 data41,15,133,254,160,0,165,252,10,10,10
1780 data10,5,251,153,0,4,153,0,5,153,0,6,153
1790 data0,7,200,208,241,165,254,10,10,10,10
1800 data5,253,153,0,68,153,0,69,153,0,70,153
1810 data0,71,200,208,241,228,197,240,252,76,35
1820 data195,120,169,49,141,20,3,169,234,141,21
1830 data3,173,0,221,9,3,141,0,221,169,255,141
1840 data13,220,169,0,141,26,208,89,96,0,0,0,0
1850 data0,173,25,208,141,25,208,173,0,221,73
1860 data1,141,0,221,76,49,234,0,0

```

```

1090 data192,50,0,52,64,53,128,54,192,55,0,57,64
1100 data58,128,59,192,60,0,62,169,0,141,0,255
1110 data169,32,133,252,169,96,133,254,160,0,132
1120 data251,132,253,152,145,251,145,253,200,208
1130 data249,230,252,230,254,165,252,201,64,208
1140 data208,169,28,133,252,169,92,133,254,173
1150 data102,19,145,251,173,103,19,145,253,200
1160 data208,243,230,252,230,254,165,252,201,32
1170 data208,233,120,169,1,141,184,3,169,3,141
1180 data189,3,169,9,160,19,32,212,139,169,4,160
1190 data19,32,46,136,32,132,140,162,63,160,19
1200 data32,0,140,169,19,160,19,32,212,139,169
1210 data14,160,19,32,46,136,32,132,140,162,68
1220 data160,19,32,0,140,172,24,19,173,25,19,32
1230 data60,121,169,63,160,19,32,73,139,162,73
1240 data160,19,32,0,140,172,26,19,173,27,19,32
1250 data60,121,169,68,160,19,32,73,139,162,78
1260 data160,19,32,0,140,169,4,160,19,32,212,139
1270 data162,38,160,19,32,0,140,169,4,160,19,32
1280 data212,139,162,38,160,19,32,0,140,169,14
1290 data160,19,32,212,139,162,48,160,19,32,0,140
1300 data169,38,160,19,32,212,139,162,43,160,19
1310 data32,0,140,169,33,160,19,32,212,139,162
1320 data58,160,19,32,0,140,162,53,160,19,32,0
1330 data140,169,0,133,250,169,53,160,19,32,212
1340 data139,169,53,160,19,32,36,138,162,83,160
1350 data19,32,0,140,162,88,160,19,32,0,140,169
1360 data58,160,19,32,212,139,169,58,160,19,32
1370 data36,138,162,93,160,19,32,0,140,169,83,160
1380 data19,32,46,136,169,43,160,19,32,69,136,162
1390 data83,160,19,32,0,140,169,58,160,19,32,212
1400 data139,169,53,160,19,32,36,138,230,99,169
1410 data48,160,19,32,69,136,162,58,160,19,32,0
1420 data40,169,83,160,19,32,212,139,162,53,160
1430 data19,32,0,140,230,250,165,250,205,3,19,240
1440 data26,169,88,160,19,32,212,139,169,93,160
1450 data19,32,69,136,169,28,160,19,32,135,140
1460 data16,3,76,160,20,173,98,19,41,7,170,208
1470 data49,173,100,19,74,74,41,254,168,185,112
1480 data19,133,251,85,113,19,133,252,173,100
1490 data19,41,7,24,101,251,133,251,144,2,230,252
1500 data24,173,98,19,41,248,101,251,133,251,165
1510 data252,109,99,19,133,252,165,250,41,1,240
1520 data9,189,104,19,160,0,17,251,145,251,165
1530 data250,41,2,240,31,165,252,73,64,133,252
1540 data169,63,141,0,255,189,104,19,160,0,17,251
1550 data145,251,169,0,141,0,255,165,252,73,64
1560 data133,252,238,98,19,208,3,238,99,19,169
1570 data38,160,19,32,212,139,169,73,160,19,32
1580 data69,136,162,38,160,19,32,0,140,173,99,19
1590 data205,25,19,240,3,76,121,20,173,98,19,205
1600 data24,19,240,3,76,121,20,169,0,141,90,19
1610 data141,99,19,238,100,19,169,14,160,19,32
1620 data212,139,169,78,160,19,32,69,136,162,14
1630 data160,19,32,0,140,173,100,19,205,26,19,240
1640 data3,76,93,20,88,96,0
2000 data120,169,168,141,20,3,169,23,141,21,3,169
2010 data0,141,18,208,173,17,208,41,127,141,17
2020 data208,88,165,213,170,201,56,208,11,230,251
2030 data165,251,41,15,133,251,24,144,44,201,59
2040 data208,11,230,252,165,252,41,15,133,252,24
2050 data144,29,201,8,208,11,230,253,165,253,41
2060 data15,133,253,24,144,14,201,60,240,65,208
2070 data202,230,254,165,254,41,15,133,254,160
2080 data0,165,252,10,10,10,5,251,153,0,28,153
2090 data0,29,153,0,30,153,0,31,200,208,241,165
2100 data254,10,10,10,5,253,153,0,92,153,0,93
2110 data153,0,94,153,0,95,200,208,241,228,213
2120 data240,252,76,25,230,169,101,141,20,3
2130 data169,250,141,21,3,173,0,221,9,3,141,0,221
2140 data88,96,0,0,0,0,173,25,208,141,25,208
2150 data173,0,221,73,1,141,0,221,76,101,250,0

```

Listato 2: Mandelbrot 128

```

10 rem : Mandelbrot in interlace
20 rem : # versione C-128 #
35 restore2000:c=0:for i=5888 to 6073:reada:
   c=c+1:pokei,a:next
37 if c<>21186 then print"errore nei data":end
40 graphicl:graphic0:scnclr:bank15
50 print"1. Carica immagine"
60 print"2. Salva immagine"
70 print"3. Calcola immagine"
80 print"4. Visualizza immagine"
85 print"5. Fine"
90 print:print"Scegli:"
100 do:getkeya$:loopwhilea$<"1"ora$>"5"
110 onval(a$)gosub200,300,400,500,600
120 run
200 print:input"Nome del file ":n$
210 load("ff1"+n$),b0,p(dec("1c00"))
220 load("ff2"+n$),b0,p(dec("5c00"))
230 goto500
300 print:input"Nome del file ":n$
310 bsave("ff1"+left$(n$,13)),b0,p(dec("1c00"))
   top(dec("3fff"))
320 bsave("ff2"+left$(n$,13)),b0,p(dec("5c00"))
   top(dec("7fff"))
330 return
400 fast
410 restore:c=0:fori=4864 to 5634:reada:c=c+a:
   pokei,a:next
420 if c<>71759 then print"errore nei data...":
   slowend
430 graphicl:sysdec("1300"):slow
500 graphicl:sysdec("1700")
510 return
600 scnclr:end
1000 data76,162,19,100,130,128,0,0,0,130,0,0,0
1010 data0,130,128,0,0,130,0,0,0,64,1,200,0
1020 data31,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1030 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1040 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1050 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1060 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1070 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1080 data34,192,35,0,37,64,38,128,39,192,40,0,42
1090 data64,43,126,44,192,45,0,47,64,48,128,49

```

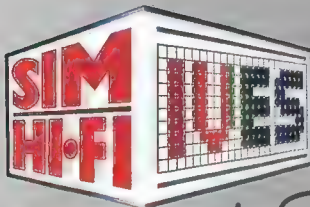
SIM-HI-FI IVES



23° salone internazionale della musica e high fidelity
international video and consumer electronics show

14-18
settembre 1989
Fiera Milano

STRUMENTI MUSICALI,
ALTA FEDELTA', HOME VIDEO,
HI-FI CAR, CAR ALARM SYSTEMS,
PERSONAL COMPUTER, TV,
VIDEOREGISTRAZIONE,
ELETTRONICA DI CONSUMO.



un grande Sim!

ASSOCIATO

HOME
VIDEO

Ingressi: Piazza Carlo Magno - Via Gattamelata - Orario: 9.00-18.00
Aperto al pubblico: 14•15•16•17 - Giornata professionale: lunedì 18 settembre

VIVA
i giovani
89

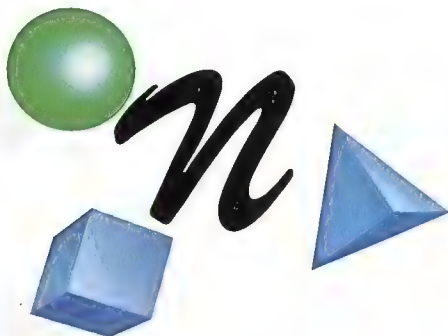


Segreteria Generale SIM-HI-FI-IVES: Via Domenichino, 11 - 20149 Milano - Tel.: 02-4815541 - Telex: 313627 - Fax 02-4980330



COMMODORE NEWS

NOVITÀ HARD E SOFTWARE DALL'ITALIA E DAL MONDO



ITALIA

LA COMMODORE CAMBIA CASA

La Commodore Italiana spa comunica che dal giorno 22 maggio 1989 il suo indirizzo è diventato:

Viale Fulvio Testi, 280

20126 Milano

tel. 02/66123.1

fax 02/66101261, telex 323496

AMIGA MEDICO

Da anni la software house Giocologico si avvale di sistemi Commodore, con particolare riguardo per i modelli della famiglia Amiga, per sviluppare applicazioni originali. L'ultima in ordine di tempo è *Amiga Medico*, uno strumento costruito per permettere al medico di base di conservare le informazioni relative ai pazienti in modo semplice e diretto. Tutte le operazioni relative alla consultazione d'archivio possono essere svolte usando il mouse. Il ricorso a immagini e simboli garantisce un accesso immediato e richiede solo un breve periodo di familiarizzazione con il programma.

Amiga Medico è scritto in linguaggio C, e l'intero progetto è basato sulle indicazioni di alcuni medici di base. La versione 1.0 del programma funziona su computer Commodore Amiga con almeno 512K di RAM con due drive da 3,5" oppure con hard disk.

Giocologico

Via Bolzano, 3

31100 Treviso

(Tel. 0422155512)

BIT MOVIE '89

Triplicate le presenze a Bit Movie '89, la seconda edizione della mostra di Computer Art organizzata a Riccione dal circolo Arci Ratataplan. Più di 1500 tra appassionati e autori hanno seguito gli stage di computergrafica, le dimostrazioni di musica elettronica, le applicazioni di desktop video e specialmente il concorso per animazione computerizzata in tempo reale che ha mostrato una notevole crescita qualitativa rispetto allo scorso anno.

Tra le 26 opere proposte da autori di tutta Italia, il pubblico ha scelto come vincitore *Atom* di Daniele Casadei (Rimini), mentre la giuria di esperti ha preferito

Moebius di Giovanni e Paolo Cignoni (Piombino). Al vincitore scelto dal pubblico è andato il premio di 500.000 lire assegnato dal Comitato Organizzatore. Apuntamento a Bit Movie '90!

Circolo Arci Ratataplan

Adriatic Coast Amiga User Club

(Tel. 0541/42878-377388)

AMIGA 1.3

Finalmente anche gli utenti italiani dell'Amiga possono avere la nuova release 1.3 del sistema operativo rivolgendosi a un Commodore Point oppure, nel caso dell'Amiga 500/2000, a un Service Center.

Amiga 1000: i tre dischi *Kickstart*, *Workbench* e *Extras* sono venduti al prezzo di lire 49.000.

Amiga 500/2000: è necessaria anche la sostituzione della ROM con quella contenente il nuovo *Kickstart*. L'intera operazione dovrebbe essere svolta in pochi minuti e il prezzo dell'integrato dovrebbe aggirarsi intorno alle 50.000 lire.

Commodore Italiana spa

Viale Fulvio Testi, 280

20126 Milano

(Tel. 02/661231)

ESTERO

FIRE KING

Con il marchio Electronic Arts, esce all'inizio dell'estate *Fire King*, un gioco realizzato dalla Strategic Studies Group.

Il grande *Fire King* è morto, e il mondo è caduto nel caos. Il giocatore viaggia attraverso foreste incantate, templi misteriosi, deserti infuocati e oscuri dungeon, risolvendo intricati enigmi e affrontando pipistrelli vampiro, vortici viventi e altri mostri, tutti pienamente animati. *Fire King* è un gioco a scroll verticale, la cui caratteristica più interessante (da un punto di vista formale) è il modo "a due giocatori" che permette a di combattere fianco a fianco o di vivere due avventure indipendenti.

Electronic Arts

11-49 Station Road, Langley
Berkshire SL3 8YN, England
(Tel. 0044/7531/46672)

DELUXE PAINT III

Finalmente è arrivato! Ormai non è più una voce di corridoio: la Electronic Arts ha annunciato ufficialmente l'uscita di *Deluxe Paint III*.

La prima versione aveva rivoluzionato il mercato, proponendo un pacchetto di disegno creativo all'altezza di programmi destinati a macchine da 20 mila dollari; la seconda era diventata un riferimento d'obbligo per chiunque volesse creare un programma di disegno per l'Amiga e ora... *Deluxe Paint III*.

Il nuovo pacchetto aggiunge al disegno l'animazione, senza togliere nulla delle opzioni fornite dalla precedente versione. È stato inserito il supporto al modo extra half-brite e si può disegnare in overscan senza problemi (in

GEOS 128 2.0

Finalmente la Berkeley Softworks ha annunciato la tanto attesa versione 2.0 di *GEOS 128*. Diciamo subito che, a parte la risoluzione di 640 x 200 pixel che sotto la guida di *GEOS* trasforma il C-128 in un Macintosh a 2 MHz, *GEOS 128 2.0* è identico all'analogica versione per il C-64, alla quale abbiamo dedicato il box "GEOS 2.0, l'avventura continua" apparso nel numero scorso di *Commodore Gazette*. Tutti i limiti che affliggevano *GEOS 128* sono stati spazzati via con un colpo di spugna: la qualità di *GEOS 128 2.0* è così elevata che gli appassionati non potranno fare a meno di aggiornarsi, mentre per i novizi è davvero venuto il momento di procurarsi le chiavi del regno. Tra l'altro, l'investimento comprende *geoWrite 2.1*, un ottimo word processor per il C-128, e il programma di disegno *geoPaint* a un prezzo decisamente interessante.

È confortante riscontrare l'impegno che la Berkeley continua a dimostrare anche per il C-128, un'ottima macchina che per quanto riguarda il software non ha mai avuto molta fortuna.

Ma veniamo al pacchetto. Il manuale è un corposo libro di 300 pagine, che lascia subito intuire la complessità del sistema. Spiega tra l'altro che le applicazioni realizzate con le versioni precedenti di *GEOS* saranno pienamente compatibili con la nuova versione, purché si utilizzi l'appropriata procedura d'installazione.

Coloro che già dispongono del sistema *GEOS 128* originale possono ottenere la nuova versione per 35 dollari (più le spese di spedizione), mentre ai nuovi acquirenti costerà 69,95 dollari. Le applicazioni incluse nel pacchetto sono le nuove versioni di *deskTop*, *geoWrite* e *geoPaint*, con in più *geoSpell*, desk accessory rinnovati, e gli interessanti paint drivers.

DeskTop è radicalmente migliorato. Il cestino si trova molto distante dall'icona della stampante, eliminando così le possibilità di cancellazioni involontarie; se comunque si dovesse ancora cancellare un file per sbaglio, il nuovo *deskTop* consente di recuperarlo. In alto a destra compare finalmente l'atteso orologio, sempre presente e soprattutto facilmente regolabile, così da dare un senso all'ora e alla data che *GEOS* associa ai file. E inoltre possibile disporre di tre disk drive, o meglio di due disk drive attivi, come di consueto, e uno in stand-by che all'occorrenza può essere reso attivo mettendone in stand-by un altro. Come nella precedente versione, possiamo impiegare indifferentemente il 1541, il 1571, il 1581, l'espansione RAM 1764 da 256K e la 1750 da 512K (almeno una delle due è sempre raccomandabile in ambito *GEOS*).

Altre interessanti novità sono la possibilità di agire su gruppi di file come su entità uniche e quindi di copiarli e cancellarli con un'unica operazione, di colorare le icone, di gestire e organizzare più facilmente le pagine delle directory sui dischi.

GeoWrite 2.1 rappresenta il punto di forza dell'intero pacchetto. Funzionando a 80 colonne, offre una completa visuale sull'intera larghezza di pagina, diventando così un vero e proprio word processor WYSIWYG. È stato notevolmente migliorato sotto il profilo della velocità e propone tutte le altre migliorie della versione 2.0 realizzata per il C-64 con in più la capacità di leggere file generati da *geoPublish*. *GeoPaint* continua a funzionare sia a 40 che a 80 colonne, ed è ora in grado di disegnare ellissi, poligonali e rette a 45 gradi. Può eseguire svariate operazioni sulle immagini scrap: ridimensionamento (proporzionale e non), distorsione, spostamento lungo lo schermo, creazione di trasparenze e così via. Inoltre mette a disposizione una griglia per muoversi su riferimenti costanti e limitare a particolari aree l'azione dei tool.

Le altre applicazioni comprese nel pacchetto sono *geoSpell*, notevolmente più veloce e funzionante solo a 80 colonne; *Text Grabber* per importare testi redatti con altri word processor per il C-128, *geoMerge* per stampare la stessa lettera più volte con sostituzione automatica di alcune parti. Segnaliamo i due paint driver, che consentono di trasformare le pagine di un testo redatto con *geoWrite* in immagini modificabili con *geoPaint*, e all'occorrenza di sovrapporre per ottenere testi disposti su più colonne e altri effetti. Anche alcuni desk accessory, come *NotePad* e *Calculator*, sono stati migliorati (possono per esempio comunicare con altre applicazioni) e la Berkeley ha inoltre aggiunto driver di stampa che possono sfruttare la maggiore densità di stampa di alcune stampanti.

Sembra che il mondo di *GEOS* sia ben lontano dall'aver esaurito le possibilità di evoluzione, e chi attende con interesse e curiosità le nuove proposte della Berkeley (anche per il C-64) difficilmente resterà deluso.

Berkeley Softworks,

2150 Shattuk Avenue,
Berkeley, CA 94704, USA (Tel. 001/415/6440883)

DPaint II era necessario lo scroll per vedere tutto il disegno). Vi sono inoltre nuovi tipi di pennello per colorare oggetti già formati e per trasferire il colore di foreground a qualunque oggetto sullo schermo (cosa che permette di colorare senza difficoltà gli oggetti in bianco e nero). Si può utilizzare l'airbrush congiuntamente a qualunque altro tool (line, circle...) realizzando sfumature più efficaci.

È stata migliorata anche la possibilità d'inserire testi, ma di tutte le novità parleremo più diffusamente nella recensione che riserveremo al programma. Spendiamo però ancora due parole sull'animazione, aiutandoci con un esempio.

Supponiamo di voler creare un'animazione in 30 fotogrammi di un F-16 che bombarda un carro armato in movimento. Si comincia disegnando lo sfondo, che verrà automaticamente ripreso in ogni fotogramma, e si passa quindi a disegnare separatamente il jet e il carro. Per muovere le immagini attraverso lo schermo basta prelevarle come si fa per i pannel-

li e portarle in giro per lo schermo tenendo premuti contemporaneamente il pulsante sinistro del mouse e un tasto dell'Amiga. In particolare, per rappresentare l'avvicinamento del jet si utilizza la procedura descritta, azionando in più i tool Perspective. Si mantiene comunque un pieno controllo su grandezza, posizione e velocità.

Deluxe Paint III offre anche un'altra entusiasmante possibilità: permette di caricare e alterare ogni file ANIM creato con altri pacchetti d'animazione, come per esempio *VideoScape*.

Deluxe Paint III è già disponibile al prezzo di 129.95 dollari, ma i possessori delle versioni precedenti dovranno inviare soltanto 50 dollari più 7 per le spese postali, e la copertina del manuale. Per chi ha acquistato *Deluxe Paint II* dopo il 1 dicembre 1988, l'aggiornamento alla nuova versione costa soltanto 7 dollari.

Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404, USA
(Tel. 0011415/5717171)

IL NUOVO DEVPAC PER L'AMIGA

La HiSoft offre una nuovissima versione del suo sistema di programmazione in linguaggio Assembly per l'Amiga, *Devpac*, che prevede un editor a tutto schermo, un macro assembler e un debugger simbolico.

La software house ha inoltre aggiunto multisection code, etichette più lunghe, etichette locali e una direttiva include binaria per l'assembler. Il nuovo assembler è tre volte più veloce del precedente e prevede la compatibilità con i codici sorgente previsti dall'assembler Commodore.

L'editor a tutto schermo rende possibile una più efficiente gestione dei blocchi e una più pratica selezione dei file. Il nuovo debugger impiega finestre multiple, permettendo al programmatore di tener d'occhio contemporaneamente più aree di memoria.

Hisoft
The Old School, Greenfield
Bedford MK45 5DE, England
(Tel. 00441525/1718181)

ERRATA CORRIGE

Nel numero 1/89, a pagina 87, nel listato 1, il ciclo FOR della settima riga dev'essere modificato come segue:

```
for (a=1 ; a<=10 ; a++)
```

Nel listato 2 della stessa pagina il ciclo FOR alla tredicesima riga dev'essere modificato come segue:

```
; for (a=1 ; a<=10 ; a++)
```

Manoscritti: le collaborazioni dei lettori - manoscritti, disegni e/o fotografie - sono benvenute e verranno valutate in vista di una possibile pubblicazione. Commodore Gazette non si assume comunque responsabilità per perdite o danni al materiale. Si prega di allegare una busta affrancata e indirizzata per ogni articolo. Il pagamento per materiale non richiesto viene effettuato solo in seguito all'accettazione da parte della redazione. I contributi editoriali (di qualunque forma) non si restituiscono. Tutta la corrispondenza editoriale, richieste di annunci, problemi di sottoscrizione abbonamenti, di diffusione e con gli inserzionisti, deve essere indirizzata a: Commodore Gazette - Uffici Editoriali - Via Monte Napoleone, 9 - 20121 Milano

Commodore Gazette è un periodico indipendente non connesso in alcun modo con la Commodore Business Machines e con tutte le sue sussidiarie e affiliate, compresa la Commodore Italiana S.p.A. Commodore Gazette viene pubblicata mensilmente dalla IHT Gruppo Editoriale, Via Monte Napoleone 9, 20121 Milano. Il costo degli abbonamenti è il seguente: Italia: 12 numeri L. 96.000, 24 numeri L. 192.000, 36 numeri L. 288.000. Estero: Europa L. 150.000 (10 numeri), Americhe, Asia... L. 200.000 (10 numeri). Nessuna parte di questa pubblicazione può essere in alcun modo riprodotta senza il permesso scritto dell'editore. La redazione si adopera per fornire la massima accuratezza negli articoli e nei listati pubblicati. Commodore Gazette non si assume responsabilità per eventuali danni dovuti a errori od omissioni.

ABBONATI!

A
CASA
TUA
UN
REGALO
OGNI
MESE

ABBONARSI
CONVIENE!

- Sicurezza di non perdere neanche un numero
- Prezzo bloccato per un anno
- Sconto del 15% sul prezzo di copertina



- Possibilità d'includere nell'abbonamento gli arretrati
- Comodità di ricevere la rivista direttamente a casa

Per chiunque sia interessato ai computer Commodore, *Commodore Gazette* è indispensabile. Nessuna rivista in Italia offre ai suoi lettori tanta qualità con recensioni hardware e software, listati, presentazioni esclusive, informazioni di ogni genere riguardanti C-64, C-128 e Amiga.

Un abbonamento a *Commodore Gazette* è il regalo più bello che possiate fare a voi stessi e agli altri... un regalo nuovo ogni mese. Ma non è finita! Risparmierete il 15% sul prezzo di copertina e potrete includere nel prezzo dell'abbonamento anche i numeri arretrati che mancano alla vostra raccolta. Resta inteso che per ogni arretrato scelto verrà spedito un numero di prossima uscita in meno. Per esempio, chi si abbona a 12 numeri a partire dal n. 1/89 e richiede 5 arretrati, riceverà 7 numeri del 1989 e 5 arretrati.

Ritagliare e spedire a: IHT Gruppo Editoriale - Via Monte Napoleone, 9 - 20121 Milano

Sì, desidero sottoscrivere un abbonamento a 12 numeri (L. 81.600) 24 numeri (L. 163.200) di **COMMODORE GAZETTE** usufruendo così dello sconto del 15% sul prezzo di copertina

Nome e cognome

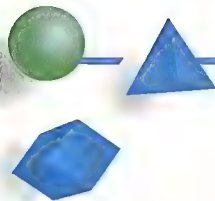
Indirizzo Città CAP

Inizio abbonamento dal n. Desidero ricevere i seguenti numeri arretrati (specificare numero e anno)

Allego assegno bancario o circolare Allego fotocopia della ricevuta del vaglia postale

Firma

Classified



Software

Compro qualsiasi programma per Litton Monroe system 2000: il s. o. è CP/M; il disco da 5 1/4 formattato ha 80 tracce. Su una pubblicità di Bit n. 53 a p. 195 c'è scritto MS/DOS compatibile; ma con quale versione dell'MS/DOS? Il microprocessore è Intel 80186. Tel. 0322/97154 dopo le 18.15 e chiedere di Marco.

Scambio programmi per C-128 sia in modo 128 che in CP/M. Annuncio sempre valido. Inviare liste a: Guido Baralla - Casella postale 20 - 55040 Ripa (LU) - Tel. 0584/760693.

Offriamo le novità per Amiga gratis! Richiediamo solo il costo del dischetto 3 1/2 vergine. Offerta del mese: dischetti 3 1/2 marcati Sony a L. 1.750, Bulk Sony a L. 1.600. Scrivere a: Stefano Maglia - Via M. dal Re, 38 - 20100 Milano. Per ulteriori informazioni: Maurizio tel. 02/2472659.

Realizziamo abbonamenti software su disco per CBM-64 a prezzo affare. Arrivi settimanali dalle più autorevoli fonti italiane ed estere. Prezzo bloccato L. 4.000 disco pieno!!! POLO SOFT - c/o Luca Pelusi - Via P. Picasso, 4 - 64023 Mosciano S. A. (TE) - Tel. 085/866280.

Svendo a L. 50.000 trattabili ben 20 dischetti pieni + manuali e la guida al C-128. Tutto per il Commodore 128 (solo in modo 128) con: *Label Maker*, *Chessmaster*, *Sex Cartoon* e altro (anche CP/M). Regalo 10 dischi vergini. Rivolgersi a: Paolo di Maio - Piazza Porta Torino, 6 - 14100 Asti - Tel. 0141/215322.

Attenzione! Saldi: causa passaggio a sistema superiore vendo in blocco 46 dischi con oltre 50 programmi per C-64/128 (C-128: *Data Manager 128*, *Superbase 128*, *DBII* (CP/M); C-64: *Tot Revolution* e tanti altri), il tutto al prezzo stracciato di L. 70.000. Per informazioni telefonare allo 0771/740118 e chiedere di Giuseppe (dalle ore 14 in poi).

Scambio programmi per Commodore Amiga. Mandare lista o telefonare a: Alfredo D'Angelo - Il trav. Livorno, 6 - 80022 Arzano (NA) - Cas. post. 134 - Tel. 081/7314158.

Scambio software per Amiga 500. Possiedo inoltre programmi MS/DOS. Gli interessati sono pregati di scrivere o telefonare a: Fabio Chiarini - Corso Roma, 138 - 20093 Cologno Monzese (MI) - Tel. 02/2543918 ore 13. Massima serietà.

Vendo in blocco 50 dischi pieni di programmi per il C-64 a L. 50.000 (compre eventuali spese di spedizione. Inoltre compro i seguenti programmi: *Microprose*

Soccer, *Neuromancer*, *Driller*, *Times of Lore*, per C-64 su disco, completi di manuali. Per maggiori informazioni scrivere a: Massimo Tabasso - Piazza Molineris, 1 - 12038 Savigliano (CN).

Cerco possessori di Amiga per scambio informazioni e software. Disponibili ultimissime news. Telefonare ore pasti e dopo le 18 a: Fabiano Martinuzzi - Tel. 0432/955594 - Majano (UD).

Scambio programmi per C-128. Se c'è qualche interessato a un Vic 20 malfunzionante (funziona in modo diretto), lo vendo per pochissimo. Adriano Locci - Località Lentisco, 9 - 57034 Campo nell'Elba (LI).

Scambio programmi per C-128, sia in modo 128 che in CP/M. Vasta biblioteca software, oltre 350 titoli, annuncio sempre valido. Inviare vostre liste a: Rosario Albanese - Corso Alberto Amedeo, 66 - 90138 Palermo - Tel. 327764.

Scambio programmi per Amiga. Contattemi, rispondo a tutti. Luca Ceria - Via

Roma, 46 - 13060 Valdengo (VC) - Tel. 015/680743 (ore serali).

Sistema Amiga. Disponibile allo scambio dei dischi di Fred Fish. Annuncio sempre valido. Solo zona di Milano. Leonardo Piccinelli - Tel. 02/3760721.

Per Amiga l'inimitabile scambio giochi, programmi e interessanti consigli. Cerco ragazzi nella zona del nord Milano per creare un gruppo di appassionati dell'Amiga. Alberto Scalco - Via dei Prati, 36 - 20030 Barlassina (MI) - Tel. 0362/561145.

Con utenti 128/64 cambio, compro, vendo tutti i migliori programmi completi di manuale originale o d'istruzioni in italiano. Massima serietà e competenza. Giuliano Cinci - Pian dei Mantellini, 44 - 53100 Siena - Tel. 0577/47054.

Vendo giochi originali ottimo stato. Telefonare a: Massimiliano - Tel. 06/3452741.

Scambio programmi Amiga anche di mia produzione: archivio prog., archivio corrispondenti, conto corrente, Invim, gestione campionato, trigonometria e molti altri; tutti a livello professionale; compilati; con mouse, gadget, menu. Scrivete a: Walter Radakovic - C. P. 2106/11 - 34100 Trieste.

Hardware

Vendo Pro 16 Steinberg originale + midi dedicata a L. 250.000 (non scindibili). Tel. 0585/52890 ore pasti. Claudio.

Vendo Amiga 1000 (512K) + monitor 1081 + drive esterno + manuali in italiano. Vendo anche separatamente, prezzo da concordare. Telefonare ore serali a Danilo Cecchin 0424/502276.

Vendo stampante MPS 1250 praticamente nuova a L. 390.000. Regalo all'acquiere pacco di mille fogli per stampante. Scrivere a: Graziano Pavone - Via B. Croce, 6 - 65016 Montesilvano (PE).

Vendo monitor a colori Philips + monitor fosfori verdi + stampante MPS 803 + digitalizzatore video in tempo reale C-64 Valerio Viscovo - Tel. 081/7332961 ore pasti.

Offertissima! Cedo Amiga 2000 (senza monitor) a L. 1.600.000 perfetto, quasi mai usato. Regalo programmi. Telefonare ore serali a Claudio: 06/7474269.

Vendo ZX Spectrum 128+2 con interfaccia dischi opus d.l + svariati programmi originali + Multiface 128 + manuali + riviste + libri + cavo stampante tutto a L. 800.000. Massimo Zanca - Via Moltarone, 6 - 28041 Arona (NO) - Tel. 0322/44503.

CLASSIFIED DELLA COMMODORE GAZETTE È UN MODO ECONOMICO PER INFORMARE LA PIÙ VASTA UTENZA COMMODORE SUI VOSTRI PRODOTTI O SERVIZI. GLI ANNUNCI NON A SCOPO DI LUCRO, INVIATI DA PRIVATI, VENGONO PUBBLICATI GRATUITAMENTE (COMPILARE L'APPOSITA SCHEDA DI SERVIZIO LETTORI).

Quote: 15.000 lire per linea, minimo 4 linee. Aggiungere 5.000 lire per ogni parola in grassetto o 50.000 lire per l'intero annuncio in grassetto.

Condizioni: pagamento anticipato. Vengono accettati assegni e vaglia postali. Gli assegni devono essere intestati a: IHT GRUPPO EDITORIALE s.r.l.

Forma: gli annunci sono soggetti all'approvazione dell'editore e devono essere scritti a macchina o in modo molto chiaro. Una linea equivale a 40 lettere, spazi tra le parole compresi. Pregasi sottolineare le parole che si intendono scrivere in grassetto. **Informazioni generali:** gli inserzionisti devono sempre specificare nome e indirizzo completo. Gli annunci appariranno nel primo numero disponibile dopo il ricevimento.

Inviare il materiale a:
IHT GRUPPO EDITORIALE
UFFICI PUBBLICITARI
VIA MONTE NAPOLEONE, 9
20121 MILANO

Attenzione: Commodore Gazette non si assume responsabilità in caso di reclami di qualunque natura da parte degli inserzionisti e/o dei lettori. Nessuna responsabilità è altresì accettata per errori e/o omissioni di qualsiasi tipo.

Per C-64/128 liquido in blocco a L. 75.000 non tratt.: penna ottica + software, Universal Cartridge 64/128, *Geos 1* originale (mai usati), *Vizawrite 64* con manuale in it. 51 pagg., *Master 64* completo con manuale 70 pagg. (valore acquisto oltre L. 200.000). Telefonare ore pasti a: Attilio 06/9022289.

Svendo Commodore 64 in buono stato + registratore + 1 joystick + interfaccia duplicatrice + tasto reset. Il tutto a sole L. 220.000. Vendo anche separatamente. Telefonare dopo le 15.00 a: Emiliano 06/5263252.

Cerco monitor monocromatico max. L. 100.000. Tratto preferibilmente zona di Biella e Vercelli. Cambio programmi per Amiga. Telefonare o scrivere a: Massimo Lovisetto - Via De Bais, 69 - 13062 Candelo (VC) - Tel. 015/538658.

Passaggio computer Amiga vendo per L. 500.000 Atari 800XL + registratore Atari 1010 + joystick + 30 cassette software originali giochi e utility + due libri Atari + corso di programmazione Atari di Accademia (22 fascicoli e 5 cassette). Marcello Bianca - Via Gela, 12 - Caltagirone (CT) - Tel. 0933/52322 (ore 13/14 e 17/18).

Causa inutilizzo, digitalizzatore audio per Amiga, mono, funzionante, attraverso la porta parallela, con tutti i più famosi programmi (es. *Audiomaster*), vendo a L. 60.000 comprese spese di spedizione. Posso fornire a richiesta anche vario software musicale. Se interessati scrivere o telefonare a: Marco Averone - Via Ribet, 9 - 10125 Torino - Tel. 011/657239.

Vendo C-128 funzionante + drive 1571 + monitor C 1901 40/80 colonne a colori + registratore. Regalo dischi, cassette e manuali. Vendo enciclopedia "Input" e tutti i numeri di *Commodore Gazette*. Telefonare a: Salvatore 0776/310619 - Via E. De Nicola - 03043 Cassino (FR).

Compro drive esterno per Amiga. Scambio e vendo programmi e manuali per Amiga. Telefonare o scrivere a: Fabio Fabbri - Via Montecuccoli, 119 - 41010 Limidi (MO) - Tel. 059/561586.

Acquisto C-64 portatile (drive e monitor incorporati) oppure SX-64 per uso viaggio. Contattare: Pietro Cremona - Casella postale 60 - 41043 Formigine (MO) - Tel. 059/570824.

Vendo per Amiga 1000 espansione di memoria interna da 1MB autoconfigurante (senza saldature), installazione semplicissima. Con *Kickstart 1.2 Ramint* e *Workbench 1.2 Ramint*. Prezzo L. 750.000. Maurizio Vignazzi - Via Michetti, 16 - 20158 Milano - Tel. 02/3760381 (ore 19.30).

Vendo Amiga 500 nuovissimo con garanzia, imballaggi, manuali in italiano, modulatore, 2 joystick, archivio programmi e varie riviste il tutto a L. 900.000 trattabili. Per informazioni rivolgersi a: Benito Babuscio - Via Principe Amedeo, 31 - 82100 Perrillo (BN) - Tel. 0824/42425.

Vendo Commodore 64 + registratore C2N + 600 programmi al prezzo di L. 220.000. Cambio anche con disk drive esterno 3,5" per Amiga. Per contattarmi scrivere o telefonare a: Carmine Evangelista - Via Santoli, 19 - 83100 Avellino - Tel. 0825/24237.

Vendo C-128 + disk drive 1571 + MPS 1200 + modem 6499 a L. 1.000.000 trattabili. Per informazioni telefonare o scrivere a: Mario Fazio - Via Roma, 306 - 00011 Guidonia (Roma) - Tel. 0774/378128.

Vendo CBM 64 + disk drive 1541, 2 registratori, 2 joystick, mouse, *Geos*, penna ottica su disco, 70 cassette piene, duplicatore cassette a L. 700.000. Valter Rivoira - Via Marucchi, 6 - 12037 Saluzzo (CN) - Tel. 0175/42529.

Vendo a prezzo ribassato scheda Janus XT (un compatibile IBM dentro Amiga 2000) ancora da acquistare, perché accessorio indesiderato facente parte di offerta Commodore (garanzia Commodore). Telefonare a: Gianni 02/2482595 o Mario 0544/31249.

Vendo C-128 + drive 1570 + stampante a colori Okimate 20 + 2 registratori + interfaccia duplicassette + cavo 40/80 colonne + joystick + foradischetti + vari libri in italiano, tutto perfettamente funzionante e con imballi originali, prezzo trattabile. Tel. 055/969288 (possibilmente zona Valdar-
no).

Varie

Vendo riviste: *MC Microcomputer* numeri 43, 44 e dal 46 al 53. *SuperVic* e *C-64* numeri 2, 3, 4, 5, 6. *Super Commodore* dal n. 1 al 7 con relative cassette. *Noi 64* e *128* i primi 3 numeri con cassette. Prezzi di copertina. Telefonare a: Antonio Di Palma 081/923810 (ore pasti e serali).

Sono disponibili le traduzioni in italiano di: *Geos 128*, *Fontmaster 128*, *Spellmaster 128*. Telefonare allo 0985/85055 dopo le ore 21.

Cerco possessori di Amiga 500 possibilmente Brescia o provincia, disposti a insegnarmi qualcosa; sono un neofita non più giovanissimo. Ermanno Dordoni Tel. 030/9547144.

Cerco appassionati di robotica applicata ai computer Commodore (C-64/128/A2000), per scambio idee. Matteo Mairate - Via Fiume, 96 - 10088 Volpiano (TO) - Tel. 011/9884461.

Cerco manuale (anche fotocopie) del Macro Assembler per C-64 solo in italiano. Telefonare a: Giuseppe Cannella - 091/6885475.

Cerco decompilatore per necessaria modifica su file compilato con il compilatore *Basic 128* della Data Becker, oppure impor-

atore dei prodotti Data Becker. Scrivere o telefonare a: Luigi Siciliano - Via Roma, 19 - 94012 Barrafranca (EN) - Tel. 0935/24161 ore ufficio.

Abbiamo fondato l'associazione "Incontriamoci" per stabilire contatti in tutta Italia tra coloro che sono alla ricerca di nuove amicizie o di persone con le quali condividere i propri hobby. A coloro che aderiranno verrà offerto un lavoro con grandi possibilità di guadagno. Contattateci Ugo Silato - Via della Resistenza, 17 - 12030 Mantia (CN).

Corrispondo con possessori Amiga scopo amicizia. Non vendo software. Scrivete a: Alfonso Roselli - Via Sannio, 32 - 65016 M. Silvano (PE).

Avete un modem e non sapete cosa farne? Bene, allora telefonate allo 090/51451 o al BBS di Messina allo 090/2929261. Parametri: baud 300, p. none, full duplex.

Cerco parti o interi del microcomputer Amico 2000, venduto in kit anni fa dalla ditta A.S.E.L. srl. Pago fino a L. 100.000 trattabili se buono. Tel. 0125/84750 (ore serali dopo le 18).

Commodore Club

A Salerno! Via Papio, 14 - Lu., me., ve. ore 19/21 il C.I.T. (Cultura - Iniziative sociali - Tempo libero) aspetta i commodoriani 64 per scambio software.

Newssoft Amiga Club contatta possessori di Amiga. Per qualunque informazione telefonare allo: 0376/619557 ore serali e festivi, e chiedere di Sergio.

ABC: Amiga Belluno Club cerca amici per informazioni, scambi culturali, non a scopo di lucro. Se sei interessato seriamente e abiti in provincia di Belluno telefonaci Mirco - Tel. 0437/86236.

Badsoft Amiga Club cerca nuovi soci interessati a questo stupendo computer. Massima serietà. Gradito contatto telefonico. Badsoft: Alessandro Badii - Via Parenzo, 36 - 57013 Rosignano Solvay (LI) - Tel. 0586/790301.

Club Amiga cerca soci: iscrizione gratuita con possibilità di numerosi giochi e utility. Per ulteriori informazioni telefonare allo: 091/6371385 (Palermo).

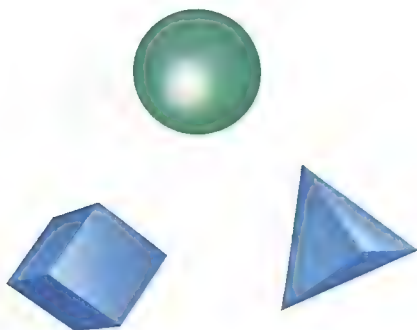
Il TNT Club Alberobello cerca nuovi soci disposti a far esplodere il loro Amiga con una miriade di programmi. Scrivere a: TNT Club - Via Montegrappa, 2 - 70011 Alberobello (BA) - Tel. 080/721871 (Giampiero).

Kiby Kenoby Club scambia programmi; nuovi arrivi ogni settimana; richiedeteci gratuitamente la lista. Scrivere a: Fausto Pasquucci - Via Galimberti, 33 - 56025 Pontedera (Pisa).

Inserzionista

Pag.

Commodore Italiana	78, 79, 80
IHT Gruppo Editoriale	1, 49, 95, III, IV
Lago	41
Newel	43
Sim Hi-Fi Ives	119
Technitron Data.....	II



Direzione vendite spazi pubblicitari:

**IHT Gruppo Editoriale
Commodore Gazette
Uffici Pubblicitari
Via Monte Napoleone, 9
20121 Milano
Tel. 02/794181 - 799492
792612 - 794122
Telex 334261 IHT I
Telefax 02/784021**

Questo indice è da considerarsi come un servizio addizionale. L'Editore non si assume alcuna responsabilità per eventuali errori e/o omissioni.







Indirizzare eventuali lamentele riguardanti gli inserzionisti a:

**Commodore Gazette
Uffici Pubblicitari
Relazioni Inserzionisti
Via Monte Napoleone, 9
20121 Milano**

Nessuna responsabilità viene altresì assunta dalla Commodore Gazette per eventuali problemi di qualsiasi natura con gli inserzionisti. La responsabilità di quanto pubblicato negli spazi pubblicitari è esclusivamente del committente.

NEL PROSSIMO NUMERO

Maggio/Giugno 1989

-  **Prove hardware** - La ricca sezione hardware comprende le prove del nuovo PC 40-III, delle schede XT e AT per l'emulazione MS-DOS e di alcune interessanti stampanti.
-  **C-64/128** - Recensioni software, listati e notizie sugli 8 bit della Commodore.
-  **Programmare l'Amiga** - Una descrizione completa di tutte le diverse temporizzazioni presenti nel sistema multitasking dell'Amiga.
-  **Amiga BASIC** - L'analisi di un pacchetto di estensione dell'Amiga BASIC e l'impiego di Intuition da Basic.
-  **Saloni USA** - Reportage dai nostri inviati negli Stati Uniti: il World of Commodore Show di Los Angeles e il Comdex '89 di Chicago.
-  **Corso di programmazione in linguaggio C** - Tutte le caratteristiche della nuova versione 5.0 del pacchetto di programmazione Lattice C.

SERVIZIO LETTORI

Questa scheda è valida fino al 30 giugno 1989

A. Come giudica questo numero di Commodore Gazette?

- 1. Ottimo
- 2. Molto buono
- 3. Buono
- 4. Discreto
- 5. Sufficiente
- 4. Mediocre
- 6. Insufficiente

B. Quale(i) articolo(i) di questo numero ha apprezzato maggiormente? _____

C. Quale(i) articolo(i) di questo numero giudica peggiore(i)? _____

D. Quali argomenti dovrebbero essere trattati nei prossimi numeri di Commodore Gazette? _____

E. Con quale aggettivo descriverebbe Commodore Gazette? _____

F. Quante persone leggono la sua copia di Commodore Gazette?

- 1. Uno
- 2. Due
- 3. Tre
- 4. Quattro o più

G. Ha dei suggerimenti? _____

H. Quale(i) computer utilizza?

- 1. C-64
- 2. C-128
- 3. C-128D
- 4. Amiga 500
- 5. Amiga 1000

- 6. Amiga 2000
- 7. PC 1
- 8. PC 10
- 9. PC 20
- 10. Altro (specificare)_____

I. Quale(i) computer intendete acquistare nel futuro?

- 1. C-64
- 2. C-128D
- 3. Amiga 500
- 4. Amiga 2000
- 5. PC 1
- 6. PC 10
- 7. PC 20
- 8. Altro (specificare)_____

L. È un acquirente dei libri della IHT? Se sì come li giudica? _____

M. Quali altre riviste (sia d'informatica che non) legge abitualmente? _____

N. Indichi quali sono i suoi maggiori interessi

- 1. Videoregistrazione
- 2. Hi-Fi
- 3. Fotografia
- 4. Automobili
- 5. Sport
- 6. Viaggi

Nome _____
Cognome _____
Indirizzo _____
Città _____
Prov. _____ C.a.p. _____
Età _____ Professione _____

COMMODORE
GAZETTE

aprile 1989



SCHEDA ORDINAZIONE LIBRI

Con il presente tagliando desidero ordinare i seguenti libri:

Collana Informatica	<input type="checkbox"/> L'Amiga (Michael Boom)	L. 60.000	
	<input type="checkbox"/> Il Manuale dell'AmigaDOS (Commodore-Amiga)	L. 60.000	
	<input type="checkbox"/> Programmare l'Amiga Vol. I (Eugene P. Mortimore)	uscita: ottobre '89	
	<input type="checkbox"/> Programmare l'Amiga Vol. II (Eugene P. Mortimore)	uscita: luglio '89	
	<input type="checkbox"/> Guida ufficiale alla programmazione di GEOS (Berkeley Softworks)	L. 64.000	
Collana Cinema	<input type="checkbox"/> Flight Simulator Co-pilot (Charles Gulick)	L. 30.000	
	<input type="checkbox"/> Volare con Flight Simulator (Charles Gulick)	L. 45.000	
	<input type="checkbox"/> Le mille luci di Hollywood (David Chell)	L. 42.000	
	Collana Tempus	<input type="checkbox"/> Inventori del nostro tempo (Kenneth A. Brown)	L. 42.000
		<input type="checkbox"/> Computer in guerra: funzioneranno? (David Bellin & Gary Chapman)	L. 39.900

Pagherò in contrassegno al postino la somma di L..... + spese postali (L. 6.000 per volume)



Nome e cognome _____
Indirizzo _____
Città _____
Prov. _____ C.a.p. _____ Tel. _____
Firma _____

COMMODORE
GAZETTE

aprile 1989

- Desidero inserire gratuitamente un mio annuncio nella rubrica CLASSIFIED (solo per i privati e per gli annunci non a scopo di lucro).
- Desidero inserire un mio annuncio nella rubrica CLASSIFIED. Allego assegno bancario o circolare o postale oppure fotocopia della ricevuta del vaglia postale per un totale di L. _____ . Il mio codice fiscale o partita IVA (per le aziende) è il seguente: _____.

Attenzione: perché un annuncio venga accettato è necessario che sia stato compilato anche il questionario presente sull'altro lato di questo tagliando. Non si accettano fotocopie.

TESTO: _____

Inserire all'interno di una busta affrancata e spedire a:

**Commodore Gazette
Servizio Lettori
Via Monte Napoleone, 9
20121 Milano**



Inserire all'interno di una busta affrancata e spedire a:

**IHT Gruppo Editoriale
Divisione Libri
Via Monte Napoleone, 9
20121 Milano**

GEOS

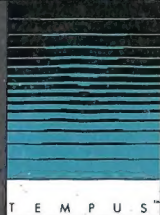
NON HA PIÙ SEGRETI



S_{ei}

un utente di GEOS e ne sei entusiasta? Desideri scoprirne ogni dettaglio e impararne il funzionamento? Vuoi programmare in questo sistema operativo ad ambiente grafico? Allora questo volume della IHT ti è indispensabile. Scritta dagli stessi creatori di GEOS, questa guida svela tutti i segreti del sistema operativo e permette di creare stupende applicazioni per il C-64 e il C-128 dotate di interfaccia utente grafica, menu, icone, finestre, box di dialogo, fonti proporzionali, processi in multi-tasking, RAM disk, gestione dei file in overlay, controllo del mouse, driver di input e di stampa, e tante altre caratteristiche che fanno di GEOS un ambiente di lavoro e sviluppo davvero professionale.

592 pagine. L. 64.000



NOVITÀ TEMPUS

Sono disponibili in libreria

i primi tre volumi della serie TEMPUS - Scienza e tecnologia, una collana che la IHT dedica alla storia, all'impatto sociale e al futuro della scienza e dell'alta tecnologia.

INVENTORI DEL NOSTRO TEMPO, di Kenneth A. Brown, raccoglie 16 interviste con alcuni dei più importanti inventori di questo secolo, gli ideatori del laser, del microprocessore, del pacemaker trapiantabile, del computer Apple II...

COMPUTER IN GUERRA: FUNZIONERANNO?, a cura di David Bellin e di Gary Chapman, è un volume scritto da scienziati e ricercatori che fanno capo all'associazione "Professionisti dell'informatica per la responsabilità sociale". Fornendo un'ampia documentazione, descrive i pericoli delle ricerche finanziate negli ultimi anni dal Pentagono (guerra automatizzata, armamenti autonomi, robot killer) e spiega perché un programma ambizioso come l'SDI non sarà mai realizzabile.

IL SEGRETO DEL SUCCESSO, di G. Ray Funkhouser e Robert R. Rothberg, analizza in ogni aspetto il fenomeno della crescita economica. Attraverso storie di clamorosi successi e di spettacolari fallimenti (come quello della Atari), spiega in quali casi la crescita aziendale è sana, e quando invece si risolverà probabilmente in un disastro.



IHT
GRUPPO
EDITORIALE

I libri IHT sono disponibili nelle migliori librerie. Per ordini diretti servirsi del modulo pubblicato a pagina 127

IHT Gruppo Editoriale - Via Monte Napoleone, 9 - 20121 Milano - Tel. 02/794181-794122 - Fax 02/784021 - Telex 334261 IHT I

Distribuzione: Messaggerie Libri - Via Giulio Carcano, 32 - 20141 Milano - Tel. 02/8438141